

Biomedical Ontologies

How to make and use them

Nigam Shah
Post-doctoral Fellow, SMI
nigam@stanford.edu

Amar Das
Assistant Professor, SMI
das@stanford.edu



Daniel Rubin
Executive Director, NCBO
drrubin@stanford.edu

Kaustubh Supekar
Doctoral Candidate
Biomedical Informatics
ksupekar@stanford.edu



Data explosion in the life sciences

- Sequence information
 - The first data type to be available in large amounts
 - Has had the maximum time to be standardized
 - FASTA format is the most popular
- Expression information
 - Recent rise in abundance
- Transcription factor binding information
 - High throughput available in yeast
- Protein-Protein interaction information
 - Relatively recent rise in availability.
 - ChIP, array based.
- Past knowledge, traditional experiments, published papers.

Copyright Stanford University 2006

2

So many biological databases, so little time

- More than 1000 different databases!
- Some biological databases:

AATDB, AceDb, ACUTS, ADB, AFDB, AGIS, AMSdb, ARR, AsDb, BBDB, BCGD, Beanref, Biolmage, BioMagResBank, BIOMDB, BLOCKS, BovGBASE, BOVMAP, BSORF, BTKbase, CANSITE, CarbBank, CARBYD, CATH, CAZY, CCDC, CD4OLbase, CGAP, ChickGBASE, Colibri, COPE, CottonDB, CSNDB, CUTG, CyanoBase, dbCFC, dbEST, dbSTS, DOBJ, DGP, DictyDb, Picty_cdb, DIP, DOGS, DOMO, DPD, DPInteract, ECDC, ECGC, EC02DBASE, EcoCyc, EcoGene, EMBL, EMD db, ENZYME, EPD, Epodb, ESTHER, FlyBase, FlyView, GCRDB, GDB, GENATLAS, Genbank, GeneCards, Genline, GenLink, GENOTK, GenProtEC, GIFTS, GPCRDB, GRAP, GRBase, gRNAsdb, GRR, GSDB, HAEMB, HAMSTERS, HEART-2DPAGE, HEXAdb, HGMD, HIBD, HICD, HIVdb, HotMoleBase, HOVERGEN, HPDB, HSC-2DPAGE, ICN, ICTVDB, IL2RGbase, IMGT, Kabat, KDNA, KEGG, Klotho, LGIC, MAD, MaizeDb, MDB, Medline, Mendel, MEROPS, MGBD, MGI, MHCPEP5 Micado, MitoDat, MITOMAP, MJDB, MmiDB, Mol-R-Us, MPDB, MRR, MutBase, MycDB, NDB, NRSUB, O-lycBase, OMIA, OMIM, OPD, ORDB, OWL, PAHdb, PatBase, PDB, PDD, Pfam, PhosphoBase, PigBase, PIR, PKR, PMD, PPDB, PRESAGE, PRINTS, ProDom, Polysis, PROSITE, PROTOMAP, RatMAP, RDP, REBASE, RGP, SBASE, SCOP, SeqAnalRef, SGD, SGP, SheepMap, Soybase, SPAD, SRNA db, SRPDB, STACK, StyGene, Sub2D, SubtilList, SWISS-2DPAGE, SWISS-3DIMAGE, SWISS-MODEL Repository, SWISS-PROT, TelDB, TGN, tmRDB, TOPS, TRANSFAC, TRR, UniGene, URNADB, VBASE, VDRR, VectorDB, WDCM, WIT, WormPep, YEPD, YPD, YPM, etc.!!!!

Copyright Stanford University 2006

3

More data is good, what's the problem?

- Too unstructured:
 - from a variety of incompatible sources
 - no standard naming convention
 - each with a custom browsing and querying mechanism
 - and poor interaction with other data sources
- Difficult to use and understand the available data, information and knowledge

Copyright Stanford University 2006

4

Ontologies to the rescue

- Ontologies provide **formal specification** of how to represent objects, concepts and relationships among them
- Ontologies provide a **shared understanding [language]** for communicating biological information
- Ontologies **overcome the semantic heterogeneity** commonly encountered in biomedical databases
- Ontologies are interpretable by humans and by computer programs.

Copyright Stanford University 2006

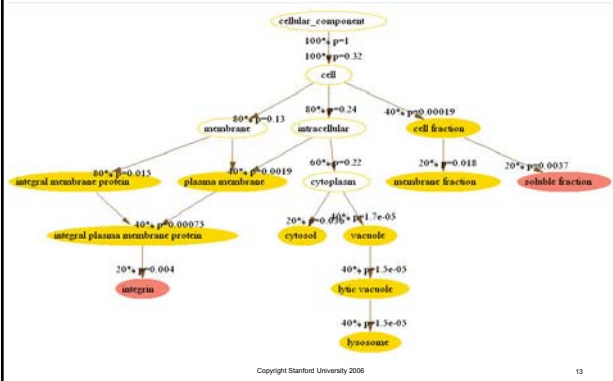
5

The National Center for Biomedical Ontology is a consortium of leading biologists, clinicians, informaticians, and ontologists who develop innovative technology and methods that allow scientists to create, disseminate, and manage biomedical information and knowledge in machine-processable form. The Center's resources include the Open Biomedical Ontologies (OBO) library, the Open Biomedical Data (OBD) repository, and tools for accessing and using this biomedical information in research. The Center collaborates with biomedical researchers conducting [Genomic Biological Research](#) (GBR) to enable their research and to stimulate technology development in the Center. The Center is undertaking [outreach and educational activities](#) to train the future generation of researchers in using biomedical ontologies and the Center's tools to enhance scientific discovery.

Copyright Stanford University 2006

6

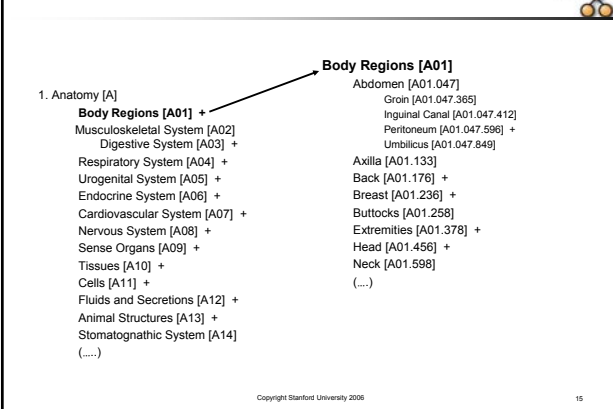
Use of GO for analysis: Shared GO terms



MESH = Medical Entity Subject Headings www.nlm.nih.gov/mesh

- Controlled vocabulary for indexing biomedical articles
- 19,000 “main headings” organized hierarchically
- Implicit semantics of parent-child relationships
- Multiple inheritance
- List of subheadings attached to main headings as modifiers

MeSH Subtrees



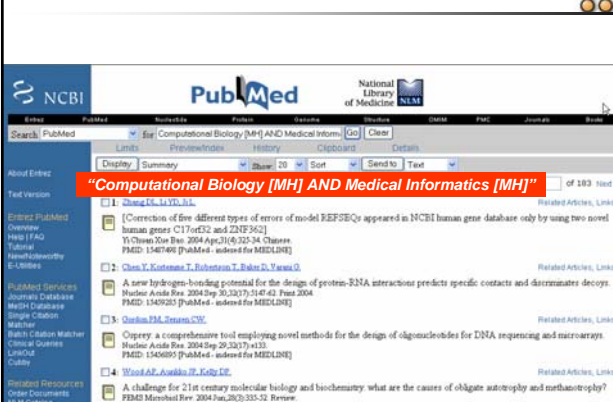
MeSH Headings in an article

MH - Adult MH - Antipsychotic
Agents/pharmacology/*therapeutic use
MH - Comparative Study
MH - Dose-Response Relationship, Drug
MH - Female
MH - Genotype
MH - Human
MH - Male
MH - Pharmacogenetics
MH - Polymorphism (Genetics)/*genetics
MH - Prognosis
MH - Psychiatric Status Rating Scales
MH - Receptors, Serotonin/drug effects/*genetics
MH - Risperidone/pharmacology/*therapeutic use
MH - Schizophrenia/diagnosis/*drug therapy/genetics
MH - Schizophrenic Psychology
MH - Support, Non-U.S. Gov't
MH - Treatment Outcome

Supplementary heading
Main headings
Minor heading
Major heading
Qualifier

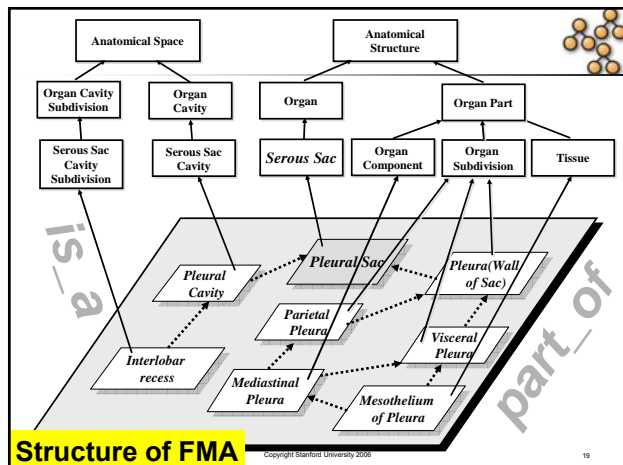
Copyright Stanford University 2006

Use of MeSH for Information Retrieval

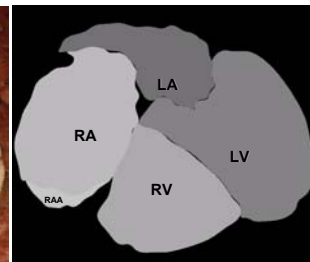
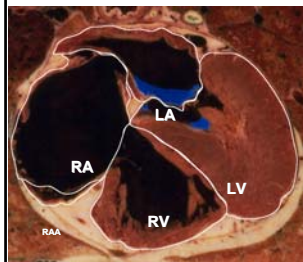


Foundational Model of Anatomy sig.biostr.washington.edu/projects/fm/

- Long-term project at University of Washington to create a comprehensive ontology of human anatomy
- 72K concepts, 1.9M relationships
- Rich semantics



Use of FMA: Image annotation



- Images possess no knowledge of their contents
- FMA-based image annotation provides that knowledge

Copyright Stanford University 2006

20

Uses of ontologies

1. Naming "things"
2. As a data exchange format
3. Define a knowledgebase schema
4. Computer reasoning over data
5. Driving NLP
6. Information integration

Copyright Stanford University 2006

21

MGED Ontology www.mged.org

- Provides **standard terms** for annotation of microarray experiments
 - Enables **unambiguous descriptions** of how the experiment was performed
 - Enables **structured queries** of elements of the experiments

Copyright Stanford University 2006

22

MGED Ontology Browser

Copyright Stanford University 2006

23

USE OF MGED ONTOLOGY: ArrayExpress Query form

Copyright Stanford University 2006

24

Uses of ontologies

1. Naming "things"
2. As a data exchange format
3. **Define a knowledgebase schema**
4. Computer reasoning over data
5. Driving NLP
6. Information integration

Copyright Stanford University 2006

25

EcoCyc

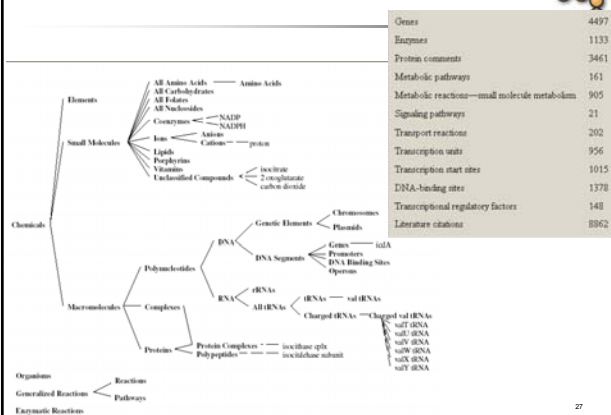
www.ecocyc.org

- The EcoCyc database is a comprehensive source of information on Escherichia coli K12.
- The mission for EcoCyc is to **contain both computable descriptions** of, and **detailed comments** describing, all genes, proteins, pathways and molecular interactions in E.coli.
- Through ongoing manual curation, extensive information has been extracted from 8862 publications and added to Version 8.5 of the EcoCyc database

Copyright Stanford University 2006

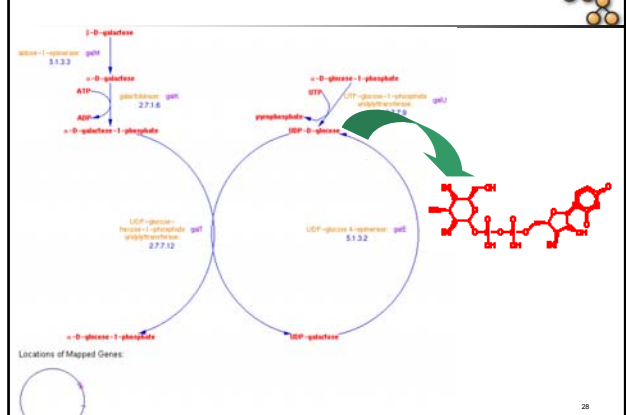
26

The EcoCyc ontology



27

Using the EcoCyc Knowledgebase



28

Uses of ontologies

1. Naming "things"
2. As a data exchange format
3. *Define a knowledgebase schema*
4. **Computer reasoning over data**
5. *Driving NLP*
6. *Information integration*

Copyright Stanford University 2006

29

Ontologies support reasoning

- Reasoning = infer new knowledge from existing assertions
- Reasoning often of two types
 - Closed world
 - Open world
- Virtual Soldier Project

Copyright Stanford University 2006

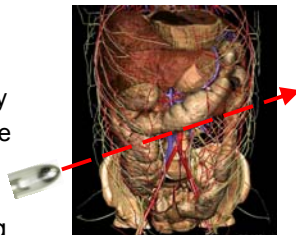
30

Our task

- Use geometric models to predict expected organ damage from penetrating injury

- Given: 3-D volumetric imaging data
- Given: injury trajectory
- Predict: organ damage and extent of injuries

This task requires anatomic reasoning



Copyright Stanford University 2006

31

Defining anatomic structures in terms of vascular supply

FMA OWL

Concept Definitions

Copyright Stanford University 2006

32

An example injury

Bullet trajectory (hitting coronary artery)

Direct damage calculated by intersecting bullet path with anatomy

Directly Injured Artery

A bullet path is described, and predicted primary injuries are displayed

Copyright Stanford University 2006

33

Inferring Injury Propagation

Totally ischemic myocardium

Partially ischemic myocardium

A computer reasoning service deduces parts of the myocardium that are at risk consequent to injury of a coronary artery, shown as highlighted structures in the ontology (above) and as shaded parts of the image of the heart (right).

Copyright Stanford University 2006

34

Uses of ontologies

- Naming "things"
- As a data exchange format
- Define a knowledgebase schema
- Computer reasoning over data
- Driving NLP**
- Information integration

Copyright Stanford University 2006

35

Geneways

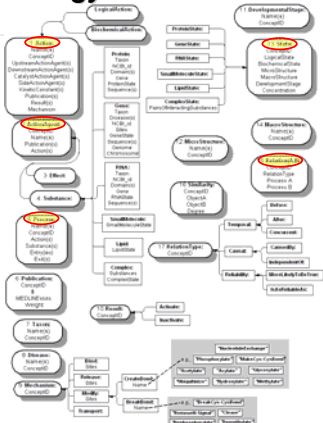
geneways.cu-genome.org

- In recent years, several groups have worked on specific problems, such as automated **selection of articles** pertinent to molecular biology, or automated **extraction of information** using natural-language processing, information visualization, and **generation of specialized knowledge bases** for molecular biology.
- GeneWays is an integrated system that combines several such subtasks. It **analyzes interactions** between molecular substances, **drawing on multiple sources of information** to infer a consensus view of molecular networks.

Copyright Stanford University 2006

36

Geneways ontology



37

Use of Geneways ontology

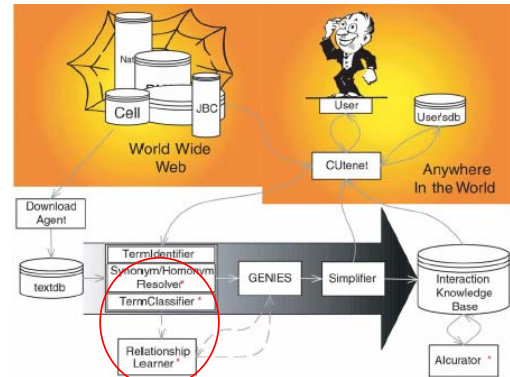


Fig. 1. A simplified view of GeneWays system.

38

Uses of ontologies

1. Naming "things"
2. As a data exchange format
3. Define a knowledgebase schema
4. Computer reasoning over data
5. Driving NLP
6. Information integration

Copyright Stanford University 2006

39

TAMBIS

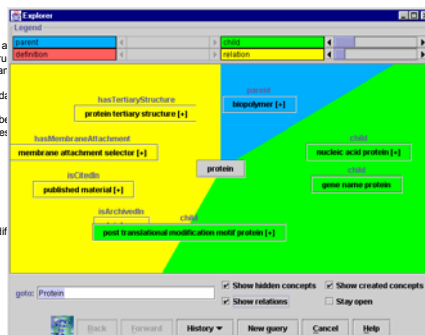
- Transparent Access to Multiple Bioinformatics Information Sources
- Motivation: Difficult to query distributed bioinformatics resources
- Concept:
 - Use an ontology to manage presentation and usage of diverse resources
 - Provide homogenizing layer over numerous heterogeneous databases & tools
 - Provide common, consistent query interface

Copyright Stanford University 2006

40

TAMBIS browser

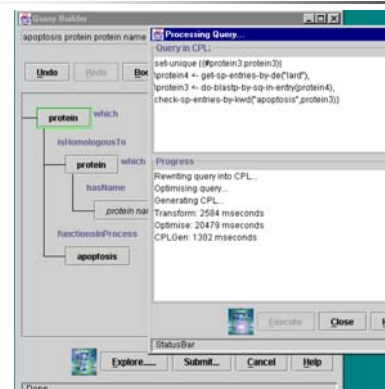
Is archived in: database
is cited in: published material
has membrane attachment: membrane
has tertiary structure: protein tertiary structure
has cellular location: organelle, membrane
has name: gene name, protein name
has secondary structure: protein secondary structure
has identifier: identifier
has accession number: accession number
functions in process: biomolecular process
is bound by: protein, binding site
binds: protein
is homologous to: protein, nucleic acid
is coded for by: exon, mRNA, DNA
is translated from: DNA, mRNA
catalyzes: reaction
has organism classification: species
has modification: post translational modification
forms part of: protein complex
has prosthetic group: prosthetic group
is expressed in organ: organ
has component: chemical binding site, post translational modification motif, domain
is component of: protein complex
is encoded by: gene
has sequence: sequence



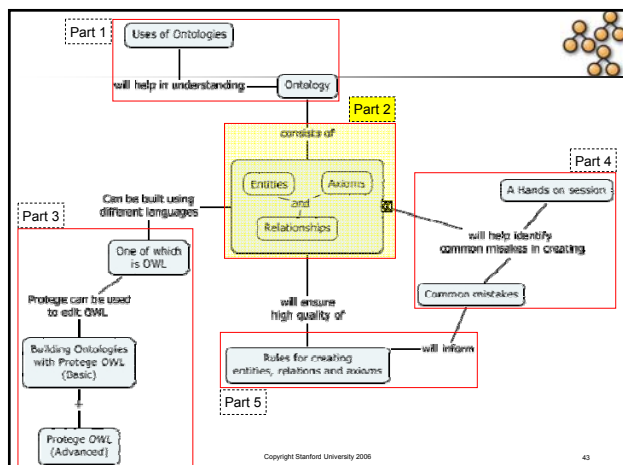
Copyright Stanford University 2006

41

Query Result



42



Various meanings of Ontology

Philosophy: Ontology is the study of what **entities** and what types of entities exist in **reality**.



AI: An ontology is an explicit specification of **concepts** & relationships that can exist in a **domain of discourse**



IT: an ontology is a **data model** that represents a domain and is used to reason about the **objects in that domain** and the relations between them

Copyright Stanford University 2006

44

The common ground...

A specification of entities (or concepts), relations, instances and axioms in an area of study.

Copyright Stanford University 2006

45

Entities

Representing entities

- | | |
|--|---|
| 1. Physical Reality | A. The reality on the side of the patient |
| 2. Psychological Reality = our knowledge and beliefs about 1. | B. Cognitive representations of this reality on the part of clinicians |
| 3. Propositions, Theories, Texts = formalizations of those ideas and beliefs | C. Publicly accessible concretizations of these cognitive representations in textual, graphical and digital artifacts |

Copyright Stanford University 2006

47

Definitions

Entity = anything which exists, including things and processes, functions and qualities, beliefs and actions, documents and software (Levels 1, 2 and 3)

Domain = a portion of reality that forms the subject-matter of a single science or technology or mode of study;

Representation = an image, idea, map, picture, name or description ... of some entity or entities.

Representational Units = terms, icons, alphanumeric identifiers ... which refer, or are intended to refer, to entities.

Copyright Stanford University 2006

48

A representation is not the same as the entity it represents



CT Scan of the Brain of Mr. X

Brain of Mr. X

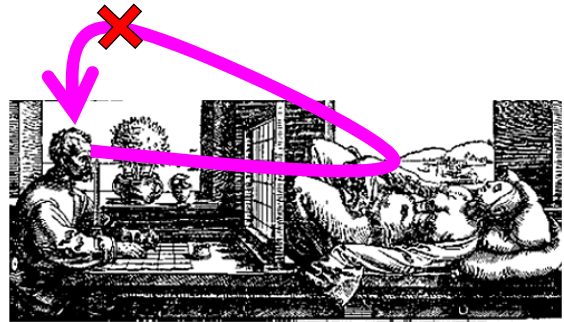
Ontology



Copyright Stanford University 2006

49

Ontologies do not represent concepts in people's heads



Copyright Stanford University 2006

50

Ontology is a tool of science

- Scientists do not describe the concepts in scientists' heads
- They describe the types (of entities and things) in reality, as a step towards finding ways to reason about (and treat) instances of these types
- An ontology is like a scientific text; it is a representation of types in reality

Copyright Stanford University 2006

51

So, an Ontology ...

- **Ontology** = a representational artifact whose representational units (which may be drawn from a natural or from some formalized language) are intended to represent
 - types in reality
 - those relations between these types which are true universally (= for all instances)

lung is_a anatomical structure
lobe of lung part_of lung

Copyright Stanford University 2006

52

Results in ...

A tension between computer scientists and philosophers.

Philosopher's view: If the Ontology is built to represent *reality* then the exchange formats and data models based on it always remains valid allowing interoperability and ... and ...

Computer scientist's view: KISS

Copyright Stanford University 2006

53

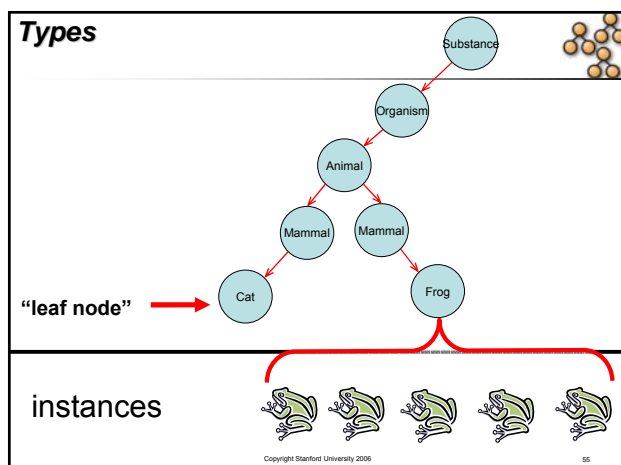
Results in the need to distinguish

Ontologies, terminologies, catalogs: represent what is general in reality = **types [classes]**

Databases, inventories: represent what is particular in reality = **instances**

Copyright Stanford University 2006

54



Classes (Types) & Defined classes (Fiat types)

Class = a maximal collection of particulars determined by a general term ('cell', 'oophorectomy' 'VA Hospital', 'breast cancer patients in VA Hospital')

- the class **A** = the collection of all particulars x for which ' x is **A**' is true

Defined Class = A class defined by a general term which does not designate a type in reality

- e.g. pathways

Copyright Stanford University 2006 56

types < defined classes < 'concepts'

- Not all of those things which people like to call 'concepts' correspond to defined classes
- “Surgical or other procedure not carried out because of patient's decision” is a concept in SNOMED ...

Copyright Stanford University 2006 57

Ontologies that represent concepts tend to make mistakes

1. congenital absent nipple **is_a** nipple concepts do not stand in
2. failure to introduce or to remove other tube or instrument **is_a** disease part_of connectedness causes treats ...
3. bacteria **causes** experimental model of disease relations to each other

Copyright Stanford University 2006 58

A Terminology is ...

A representational artifact whose representational units are natural language terms (with IDs, synonyms, comments, etc.) which are intended to represent defined classes.

Most Medical “Ontologies” are terminologies

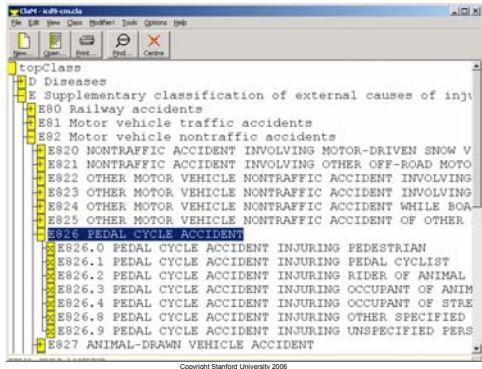
Copyright Stanford University 2006 59

The International Classification of Diseases

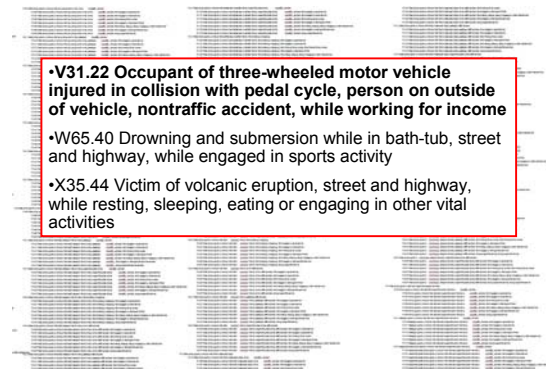
724	Unspecified disorders of the back
724.0	Spinal stenosis, other than cervical
724.00	Spinal stenosis, unspecified region
724.01	Spinal stenosis, thoracic region
724.02	Spinal stenosis, lumbar region
724.09	Spinal stenosis, other
724.1	Pain in thoracic spine
724.2	Lumbago
724.3	Sciatica
724.4	Thoracic or lumbosacral neuritis
724.5	Backache, unspecified
724.6	Disorders of sacrum
724.7	Disorders of coccyx
724.70	Unspecified disorder of coccyx
724.71	Hypermobility of coccyx
724.71	Coccygodynia
724.8	Other symptoms referable to back
724.9	Other unspecified back disorders

Copyright Stanford University 2006 60

ICD9 (1977): A Handful of Codes for Traffic Accidents



ICD10 (1999): 587 codes for such accidents



Relationships

The “is_a” relation

- What does *A is_a B* mean?
- For all *x*, if *x* **instance_of** *A* then *x* **instance_of** some *B*
- *cell division is_a biological process*

ALL-SOME STRUCTURE

The “part_of” (vs. has_part) relation

- Human being has_part testis? *A part_of B* = all instances of *A* are **instance-level parts** of some instance of *B*
- human testis part_of human being ?
- Human being has_part heart? *human testis part_of human being*
- human heart part_of human being ?

Two kinds of parthood

between instances:

Mary’s heart **part_of** Mary
this nucleus **part_of** this cell

between types

human heart part_of human
cell nucleus part_of cell

The “part_of” relation

- What does *A part_of B* mean?
- For all x , if x **instance_of** A then there is some y , y **instance_of** B and x **part_of** y
 - where ‘part_of’ is the instance-level part relation
- *cell nucleus part_of cell*

ALL-SOME STRUCTURE

Copyright Stanford University 2006

67

A part_of B, B part_of C ...

The **all-some** structure of the definitions allows cascading of inferences

1. within ontologies
2. between ontologies
3. between ontologies and EHR repositories of instance-data

Copyright Stanford University 2006

68

Adjacent_to is also of two kinds

- Instance level
 - this nucleus is adjacent to this cytoplasm implies: this cytoplasm is adjacent to this nucleus
- Type level
 - nucleus adjacent_to cytoplasm
 - Not: cytoplasm adjacent_to nucleus
 - (because you can have enucleated cells)

Copyright Stanford University 2006

69

Mathematical properties matter ...

- Expectations of symmetry e.g. for protein-protein interactions may hold only at the instance level
 - if A interacts with B , it does not follow that B interacts with A
 - if A is expressed simultaneously with B , it does not follow that B is expressed simultaneously with A

Properties of Relations

1. Transitivity
2. Symmetry
3. Reflexivity
4. Anti-Symmetry
5. ...

Copyright Stanford University 2006

70

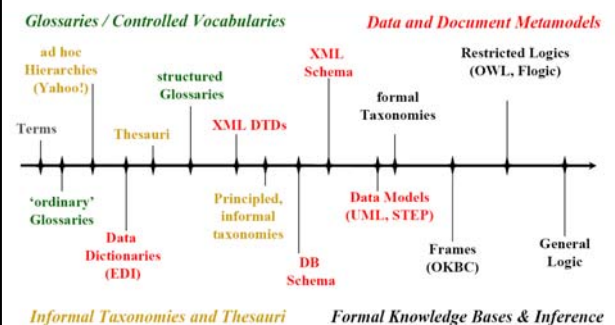
Other Ontology-like things

- **Controlled vocabulary** = A list of explicitly enumerated unambiguous terms; Controlled by a central registration authority;
- **Taxonomy** = collection of controlled vocabulary terms organized into a hierarchy
- **Thesaurus** = Collection of controlled vocabulary terms organized into a specialized network

Copyright Stanford University 2006

71

Increasing “formality”...



Originally by Michael Uschold, with permission

Copyright Stanford University 2006

72

Application vs. Reference Ontologies

- A reference ontology is analogous to a scientific theory.
 - ... consists of representations of biological reality which are correct according to our current understanding.
- An application ontology is a software artifact:
 - ...for, structuring data according to some hierarchy of classes, for the purpose of managing and manipulating that data, supporting interoperability of various resources.
- As far as possible, we should focus on developing [scientific] information models, data-models, process-models etc to be as close as possible to and refer to *reference ontologies*.

Copyright Stanford University 2006

73

Languages [formalisms] for Ontologies

- There are numerous ways of declaring both reference and application ontologies
- Almost all ontology languages give you the ability [and syntax] for declaring entities and relationships
- The main differences are in the ability [and mechanism] of describing the attributes of the entities and the mathematical properties of the relationships.
 - <http://xml.coverpages.org/OntologyExchange.html>
- Another major factor is the level of tool support available for “writing” in that language.
 - http://xml.com/2002/11/06/Ontology_Editor_Survey.html

Copyright Stanford University 2006

74

A partial list of ontology languages

1. **KIF** = Knowledge Interchange format
2. **OKBC** = Open Knowledge Base Connectivity
 - The Generic Frame Protocol is the implicit formalism underlying OKBC.
3. **OBO** = Open Biomedical Ontology
4. **OWL** = Web Ontology Language
 - Will be discussed in today's tutorial
 - Subsumes XML, RDF(S), DAML+OIL

Copyright Stanford University 2006

75

What an Ontology is NOT

- An ontology **is not** the same as a **knowledgebase**
 - Ontology (types) + Instances = KB
- An ontology **is not** the same as a **database schema**
 - A database schema is designed to store the instances conforming to an ontology
- An ontology **is not** the same as an **XSD**
 - An XSD tells you *how* to store the information that describes the instances

Copyright Stanford University 2006

76

Database Schema vs. Ontologies

Language Expressivity: Much overlap, some differences

Overlap: objects, properties, aggregation, generalization, set-valued properties, constraints

Different terms:

- Entities vs. Classes or Concepts
- Attributes and Relations vs. Relations and Properties
- Constraints vs. Axioms

Some differences, due to different purposes

- Structuring a database vs.
- Knowledge sharing/reuse, search, interoperability, specification

Originally by Michael Uschold, with permission

Copyright Stanford University 2006

77

Database Schema vs. Ontologies

Language Expressivity: Constraints

- **DB: Purpose of Constraints is**
 - **Primarily** to ensure data integrity
 - Also used to optimize queries
 - Cardinality constraints: some highly DB-specific uses
 - May also shed light on meaning
- **Ont: Purpose of Axioms is**
 - **Primarily** to express machine-readable meaning to support automated reasoning
 - May also ensure data integrity

Originally by Michael Uschold, with permission

Copyright Stanford University 2006

78

Database Schema vs. Ontologies

Systems that Implement DBS's and Ontologies

Processing engines for handling complex logical expressions:

DB: SQL engines specialized for querying, views and data integrity
Instances are fundamental vs.

Ont: Logic-based theorem provers to support automated inference

- To infer new information
- To ensure consistency

Instances

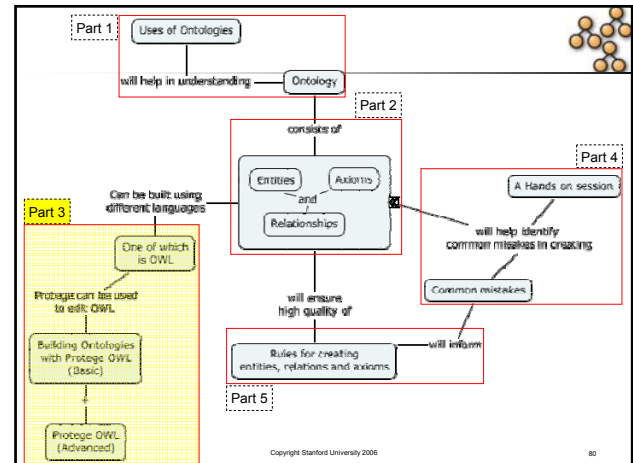
DB: the whole point

Ont: useful w/ or w/out instances

Originally by Michael Uschold, with permission

Copyright Stanford University 2006

79



Copyright Stanford University 2006

80

INTRODUCTION TO OWL

Copyright Stanford University 2006

81

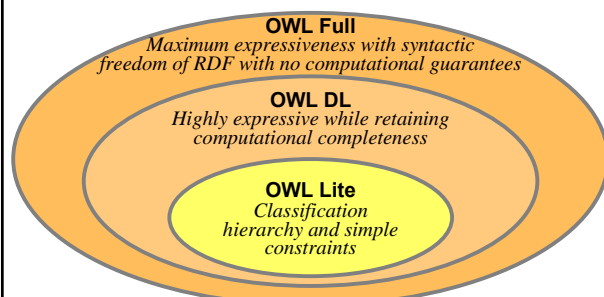
OWL

- Web Ontology Language
- Recommended by W3C since Feb 2004
- Based on predecessors (DAML+OIL)
- A Web Language: Based on RDF(S)
- An Ontology Language: Based on logic
- Three varieties
 - OWL-full
 - OWL-DL ("OWL")
 - OWL-Lite

Copyright Stanford University 2006

82

The Three Sublanguages of OWL



Copyright Stanford University 2006

83

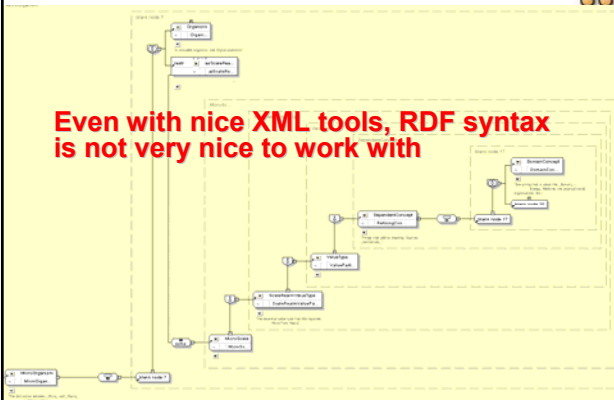
Working with OWL syntax is not easy

```
<owl:Class rdf:ID="Virus">
  <rdf:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    </rdf:comment>
  <owl:disjointWith>
    <owl:Class rdf:ID="Bacterium"/>
  </owl:disjointWith>
  <rdf:subClassOf>
    <owl:Class rdf:ID="MicroOrganism"/>
  </rdf:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Bacterium">
    <rdf:subClassOf>
      <owl:Class rdf:ID="MicroOrganism"/>
    </rdf:subClassOf>
    <rdf:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      Bacterium</rdf:label>
    <rdf:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      </rdf:comment>
  </owl:Class>
  <owl:Class rdf:ID="MicroOrganism">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:resource="Collection">
          <owl:Class rdf:ID="Organism"/>
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="MicroScale"/>
            </owl:someValuesFrom>
          </owl:Restriction>
          <owl:ObjectProperty rdf:ID="asScaleRealm"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
```

84

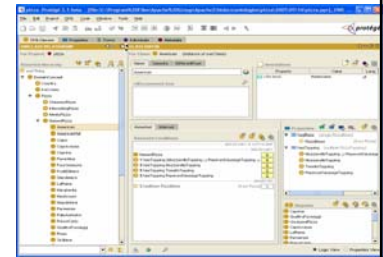
Tools are being developed for OWL

Even with nice XML tools, RDF syntax is not very nice to work with



Protégé OWL: a GUI environment for OWL

- Robust OWL environment within PROTÉGÉ framework
- Most widely used tool for editing and managing OWL ontologies



Copyright Stanford University 2006

88

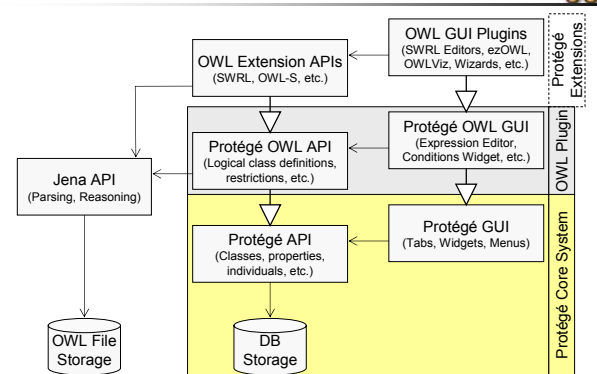
Protégé OWL features

- Loading and saving OWL files & databases
- Graphical editors for class expressions
- Access to description logics (DL) reasoners via Protégé GUI
- Ontology visualization components
- Built on Protégé platform
 - Can hook in custom-tailored components
 - API for new applications

Copyright Stanford University 2006

87

OWL Plugin Architecture



Copyright Stanford University 2006

88

Outline

- Background on OWL
- ➔ • Basic Protégé-OWL Usage
 - Projects
 - Classes
 - Properties
 - Restrictions
 - Individuals
- Classification
- Visualization components

Copyright Stanford University 2006

89

OWL PROJECTS

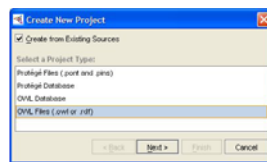
Copyright Stanford University 2006

90

Loading OWL files

1. If you only have an OWL file:

- **File** → **New Project**
- Select **OWL Files** as the type
- Tick **Create from existing sources**
- **Next** to select the .owl file



2. If you've got a valid project file*:

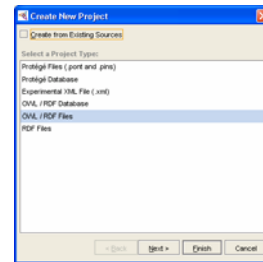
- **File** → **Open Project**
- select the .pprj file

* ie one created on this version of Protégé - the s/w gets updated once every few days, so don't count on it unless you've created it recently - safest to build from the .owl file if in doubt

Copyright Stanford University 2006

91

(Create or load an OWL project)



File → **New Project**

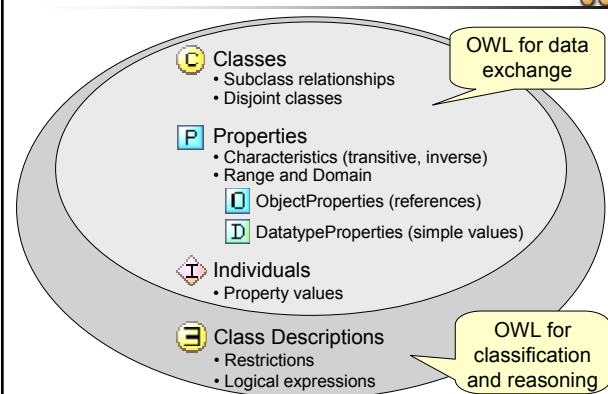
OR

File → **Open Project**

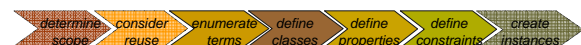
Copyright Stanford University 2006

92

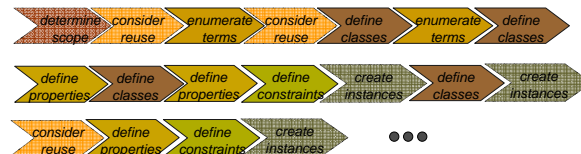
Protégé OWL Overview



Ontology Development Process



In reality - an iterative process:



Copyright Stanford University 2006

94

Establish Purpose



What will the ontology be used for?

Classification of Pneumonia:

- Bacterial Pneumonia (caused by bacteria)
- Pneumococcal Pneumonia (caused by a particular kind of bacteria)
- Viral Pneumonia (caused by viruses)
- Mixed Pneumonia (caused by both bacteria and viruses)

Copyright Stanford University 2006

95

Enumerate Important Concepts



- What are the terms we need to talk about?
–Pneumonias, infectious organisms.
- What are the properties of these terms?
–hasRadiologyFinding, hasLocus, hasCause.
- What do we want to say about the terms?
–Pneumonias cause radiology opacity findings
–Pneumonias are located in lung
–Mixed pneumonias are caused by bacteria and viruses.

...

Copyright Stanford University 2006

96

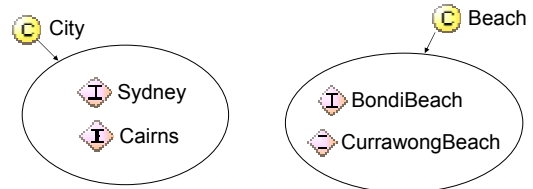
CLASSES

Copyright Stanford University 2006

97

Classes

- Sets of **individuals** with common characteristics
- Individuals** are *instances* of at least one class

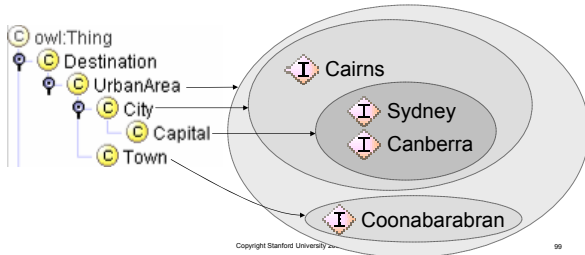


Copyright Stanford University 2006

98

Superclass Relationships

- Classes organized in a hierarchy implies subsumption
- Direct instances of subclass are also (indirect) instances of superclasses

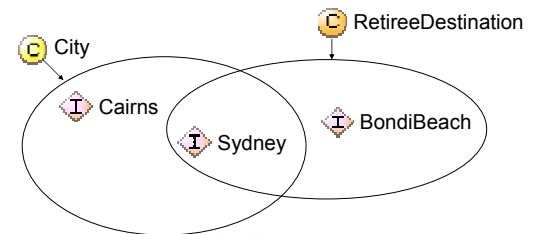


Copyright Stanford University 2006

99

Class Relationships

- Classes can overlap arbitrarily
- Classes are assumed non-disjoint by default (ie, they may share instances)

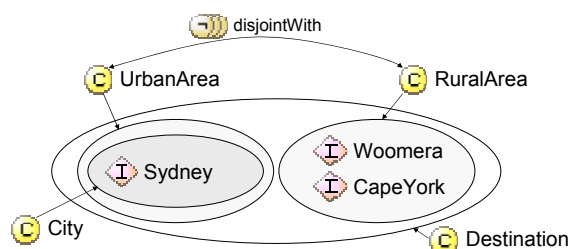


Copyright Stanford University 2006

100

Class Disjointness

- All classes could potentially overlap
- Specify **disjointness** to make sure they don't share instances

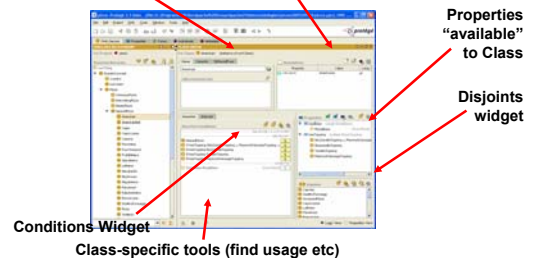


Copyright Stanford University 2006

101

Class Editor

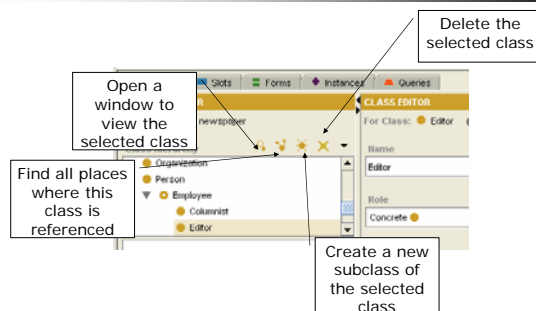
Class annotations (for class metadata)
Class name and documentation



Copyright Stanford University 2006

102

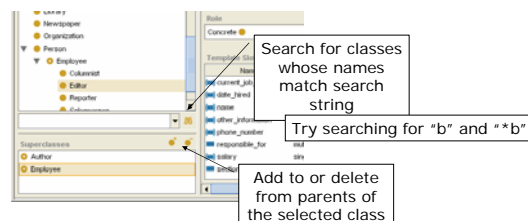
Operations on classes



Copyright Stanford University 2006

103

More operations on a class

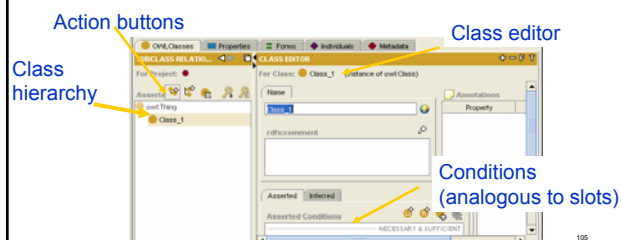


Copyright Stanford University 2006

104

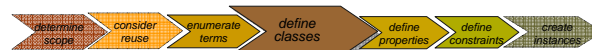
OWL Graphical User Interface

- Overall organization similar to frame-based Protégé – tabbed UI, class hierarchy, class editor, etc.
- e.g., Click the “new subclass” icon in the classes tab and create a new class



105

Define classes and the class hierarchy



- Identify Classes (from the previous term list)

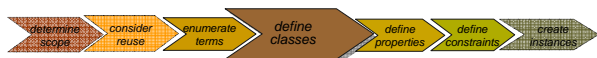
– If something can have a kind then it is a Class

- “Kind of Pneumonia” ✓ - Pneumonia is a Class
- “Kind of Samson” ✗ - Samson is an individual
- “Kind of Bacteria” ✓ Bacteria is a Class

Copyright Stanford University 2006

106

Define classes and the class hierarchy



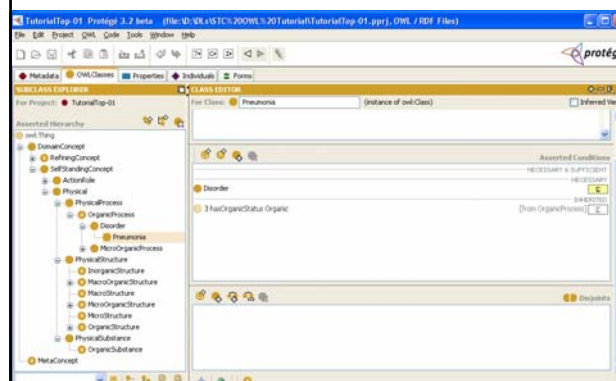
Arrange Classes in an hierarchy

- PneumococcalPneumonia is a subclass of Pneumonia
 - Every PneumococcalPneumonia is a Pneumonia
- Pneumococcus is a subclass of Bacteria
 - Every Pneumococcus is a Bacteria
- MixedPneumonia is a subclass of Pneumonia
 - Every MixedPneumonia is a Pneumonia

Copyright Stanford University 2006

107

Create classes: create “Pneumonia” class



Class Disjoints

Note that Bacterial Pneumonia

- has superclass Pneumonia as a necessary condition
- Is asserted to be disjoint from its 'siblings'

The screenshot shows the Protege interface with the 'CLASS EDITOR' window. The 'Pneumonia' class is selected, and its subclasses, 'BacterialPneumonia' and 'ViralPneumonia', are listed. A red arrow points to 'Pneumonia' with the label 'Necessary parent'. Another red arrow points to the 'Disjoint classes' section, which lists 'BacterialPneumonia' and 'ViralPneumonia' as disjoint from each other.

What it means

- All BacterialPneumonias are Pneumonias
 - No BacterialPneumonia is not a Pneumonia
- Nothing is both:
 - a BacterialPneumonia and a ViralPneumonia
 - a BacterialPneumonia and a MixedPneumonia

NB: In OWL classes **can overlap** unless declared disjoint!

Copyright Stanford University 2006

110

Add Annotations on Classes

The screenshot shows the Protege interface with the 'CLASS EDITOR' window. The 'Pneumonia' class is selected, and its subclasses, 'BacterialPneumonia' and 'ViralPneumonia', are listed. A red arrow points to the 'Pneumonia' class with the label 'Necessary parent'. Another red arrow points to the 'Disjoint classes' section, which lists 'BacterialPneumonia' and 'ViralPneumonia' as disjoint from each other.

Another Way to Create Classes

- A class can be the **union** of two classes
 - An InfectiousPneumonia is either a BacterialPneumonia or a ViralPneumonia
- A class can be the **intersection** of two classes
 - A MixedPneumonia is any Pneumonia that is caused by both Bacteria and Viruses
- A class can be the **complement** of another class
 - Noninfectious pneumonia is any pneumonia that is not caused by an infectious agent (bacteria or virus)

Copyright Stanford University 2006

112

Create a class by composition

The screenshot shows the Protege interface with the 'CLASS EDITOR' window. The 'InfectiousPneumonia' class is selected, and its subclasses, 'BacterialPneumonia' and 'ViralPneumonia', are listed. A red arrow points to 'Pneumonia' with the label 'Necessary parent'. Another red arrow points to the 'Disjoint classes' section, which lists 'BacterialPneumonia' and 'ViralPneumonia' as disjoint from each other.

An InfectiousPneumonia is a Pneumonia that is either a BacterialPneumonia or a ViralPneumonia

PROPERTIES

Copyright Stanford University 2006

114

OWL Properties


- **Datatype Property** – relates Individuals to data (int, string, float etc)
 - *Pneumonia hasRadiologyFinding xsd:String*
- **Object Property** – relates Individuals
 - *BacterialPneumonia hasCause Bacterium*
- **Annotation Property** – for attaching metadata to classes, individuals or properties
 - *OntologyClass hasAuthor Natasha*

Copyright Stanford University 2006

115

Datatype Properties

- Link individuals to primitive values (integers, floats, strings, booleans etc)
- Often: AnnotationProperties without formal “meaning”

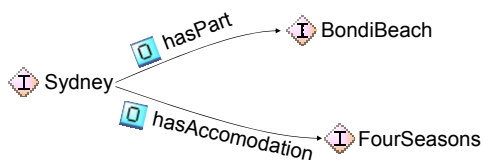
 Sydney
hasSize = 4,500,000
isCapital = true
rdfs:comment = “Don’t miss the opera house”

Copyright Stanford University 2006

116

Object Properties

- Link two individuals together
- Relationships (0..n, n..m)

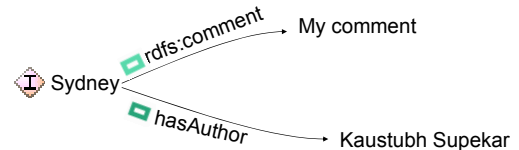


Copyright Stanford University 2006

117

Annotation Properties

- To annotate classes, properties, and individuals
- Usually used for documentation



Copyright Stanford University 2006

118

Properties of an OWL property

- Functional
 - *Person has_Mother Mother*
- Transitive
 - *A hasPart B, B hasPart C ==> A hasPart C*
- InverseFunctional
 - *Person has_SSN SSN*
- Symmetric
 - *A worksWith B ==> B worksWith A*

Copyright Stanford University 2006

119

Inverse Properties

- Represent bidirectional relationships
- Adding a value to one property also adds a value to the inverse property

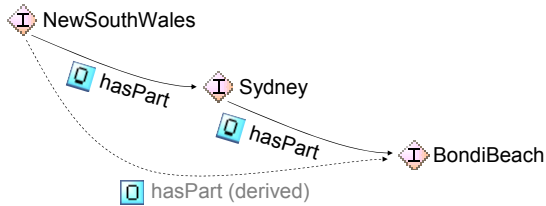


Copyright Stanford University 2006

120

Transitive Properties

- If A is related to B and B is related to C then A is also related to C
- Often used for part-of relationships

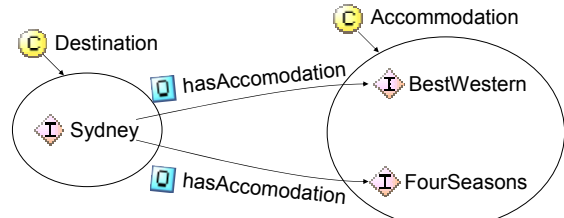


Copyright Stanford University 2006

121

Range and Domain

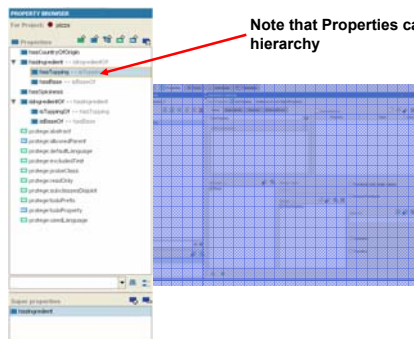
- Property characteristics
 - Domain: “left side of relation” (Destination)
 - Range: “right side” (Accommodation)



Copyright Stanford University 2006

122

Properties Tab: Property Browser



Copyright Stanford University 2006

123

Define Properties of Classes



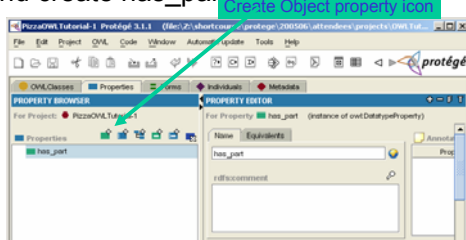
- Properties in a class definition describe attributes of instances of the class and relations to other instances
 - Each Pneumonia will have radiology findings and a cause
 - Each cause for pneumonia will have a causative organism.

Copyright Stanford University 2006

124

Create object property “has_part”

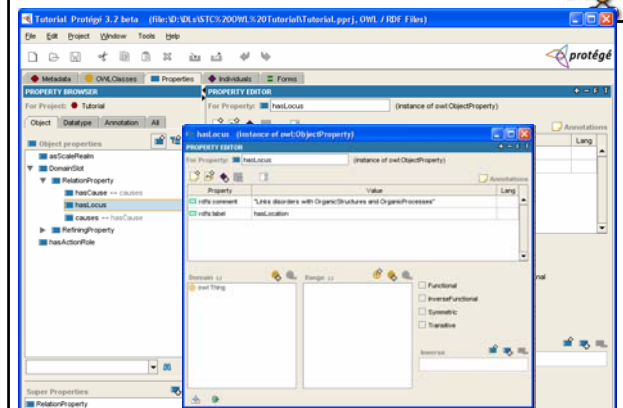
- Click on properties tab
- Click on Create_Object_property icon and create has_part



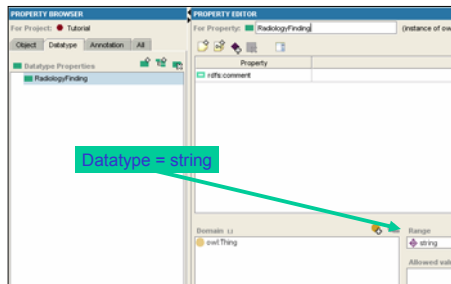
Copyright Stanford University 2006

125

Object property hasLocus (already present)



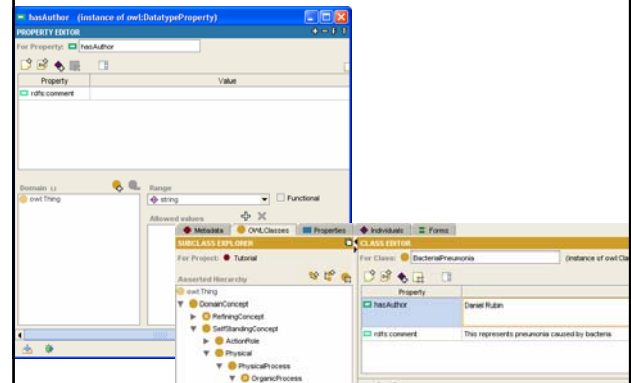
Create New Datatype Property, "hasRadiologyFinding"



Copyright Stanford University 2006

127

Create annotation property "hasAuthor"



Build a Simple Property Hierarchy



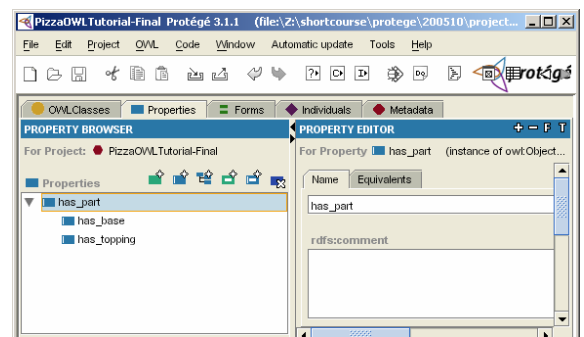
- If a pizza **has a topping**, then that topping is a **part** of the pizza
- If a pizza **has a base**, then that base is a **part** of the pizza

has_topping and has_base are subproperties of has_part

Copyright Stanford University 2006

129

Making subproperties



Break for 15 min

RESTRICTIONS

Copyright Stanford University 2006

132

Restrictions (Overview)

- An **anonymous class** consisting of all individuals that fulfill the condition
- Define a condition for property values
 - allValuesFrom
 - someValuesFrom
 - hasValue
 - minCardinality
 - maxCardinality
 - cardinality

Copyright Stanford University 2006

133

Define Constraints : OWL Restrictions



- Quantifier restriction**
 - How to represent the fact that every pneumonia must be located in a lung?
- Cardinality restrictions**
 - How to represent that a Hand must have 5 fingers as parts?
- hasValue restrictions**
 - How to define the value of a relation for a class? (relationship between class and an individual)

Copyright Stanford University 2006

134

Quantifier Restrictions

Restrictions are of the form

All members of class C have as values for property p
some things of Class D (\exists)
only things of class D (\forall)
 at least | at most | exactly n things

Examples

- “**some**” (someValuesFrom) (\exists) (**Existential**)
 Cheesy_Pizza has_base someValuesFrom Cheese_Topping.
 Implies - “All cheesy pizzas have some (at least 1) topping that is a cheesy topping”
- “**only**” (allValuesFrom) (\forall) (**Universal**)
 VegetarianPizza has_topping allValuesFrom Vegetarian_Topping.
 Implies - “All Vegetarian pizzas have only toppings that are Vegetarian Toppings”

Copyright Stanford University 2006

135

Creating Restrictions

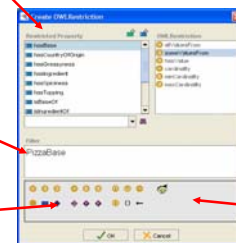
Restricted Property

Filler Expression

Expression Construct Palette

Restriction Type

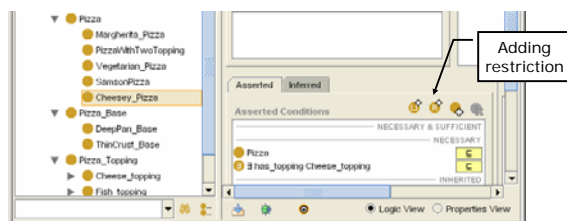
Syntax check



Copyright Stanford University 2006

136

Adding a Qualifier Restriction: Example



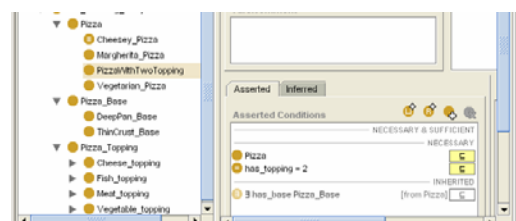
- All Cheesy_Pizzas have **some** Cheese_topping
 - \exists means “some”
 - an “existential restriction”

Copyright Stanford University 2006

137

Adding a Cardinality Restriction

- PizzaWithTwoTopping
 Pizza (hasTopping = 2)



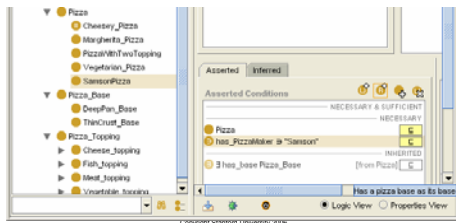
Copyright Stanford University 2006

138

hasValue Restriction

Sometimes we need to define a Class that has a property which takes individuals as values

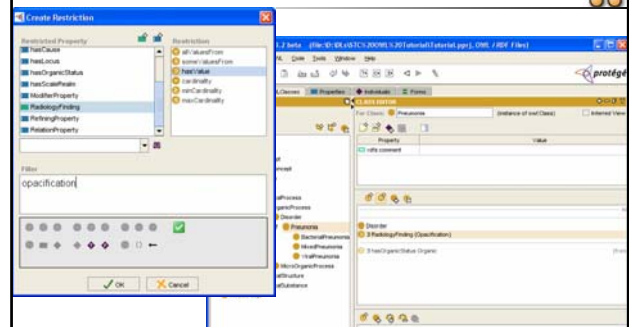
–Pizza has_PizzaMaker Samson



Copyright Stanford University 2006

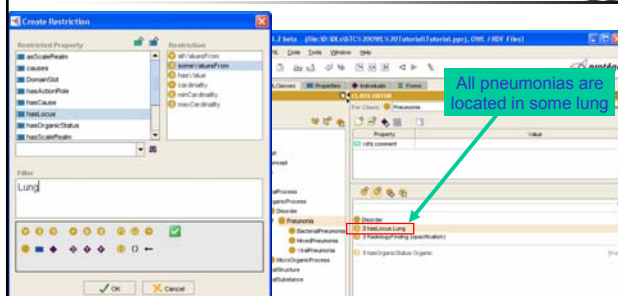
139

Create a restriction: Add a datatype property



"All pneumonias are disorders that have a radiological finding of opacification"

Add an Object Property



"All pneumonias are disorders that are located in some lung and have a radiological finding of opacification"

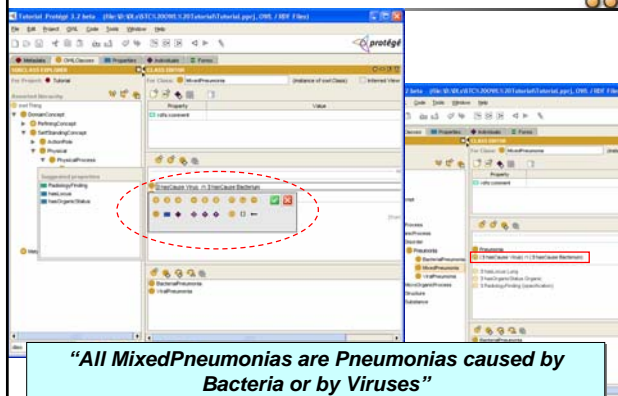
Add more object properties

- BacterialPneumonia is caused by some bacteria
 - BacterialPneumonia \sqsubseteq causedBy some Bacteria
 - BacterialPneumonia $\rightarrow \exists$ causedBy.Bacteria
- ViralPneumonia is caused by some virus
 - ViralPneumonia \sqsubseteq causedBy some Virus
- MixedPneumonia is caused by some bacteria and by some virus
 - MixedPneumonia \sqsubseteq (causedBy some Bacteria) \sqcap (causedBy some Virus)

Copyright Stanford University 2006

142

Using expression editor



"All MixedPneumonias are Pneumonias caused by Bacteria or by Viruses"

Class Descriptions

- Define the "meaning" of classes
- Description Logic expressions ("anonymous class expressions") are used:
 - "All national parks have campgrounds."
 - "A backpackers destination is a destination that has budget accommodation and offers sports or adventure activities."
- Expressions usually restrict property values
- Reasoners can perform inference/classification



144

Defined/Primitive Classes

Necessary Conditions:
(Primitive / partial classes)
"If we know that something is a X,
then it must fulfill the conditions..."

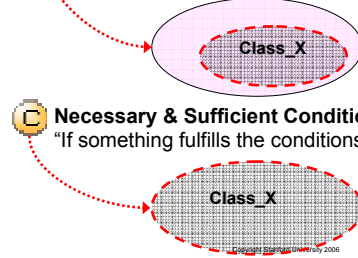
Necessary & Sufficient Conditions:
(Defined / complete classes)
"If something fulfills the conditions...,
then it is an X."



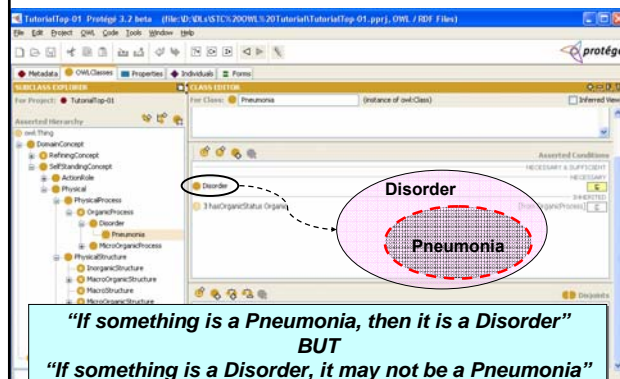
Defined/Primitive Classes

Necessary Conditions: (Primitive classes)
Describes a subclass
"If something is a Class_X, then it must fulfill the conditions..."
Converse may NOT be true: "If something fulfills the conditions...,
then it is a Class_X."

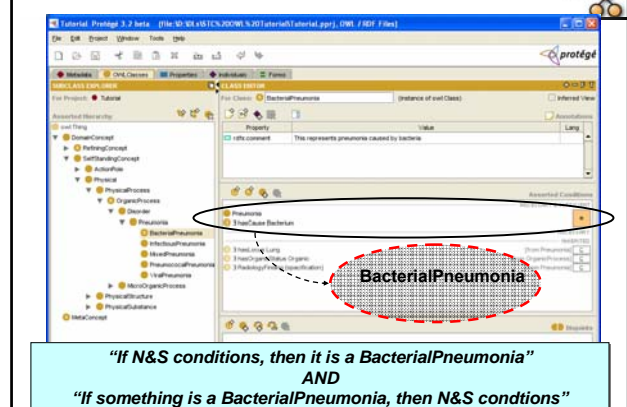
Necessary & Sufficient Conditions: (Defined classes)
"If something fulfills the conditions..., then it is a Class_X."



e.g., Disorder is a necessary condition on Pneumonia



Necessary & sufficient conditions on BacterialPneumonia



Logical Class Definitions

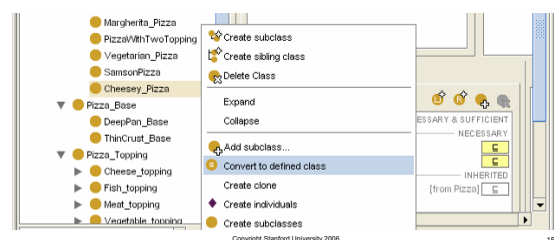
- Define classes out of other classes
 - unionOf (or)
 - intersectionOf (and)
 - complementOf (not)
- Allow arbitrary nesting of class descriptions (A and (B or C) and not D)

Copyright Stanford University 2006

149

Making Classes "Defined"

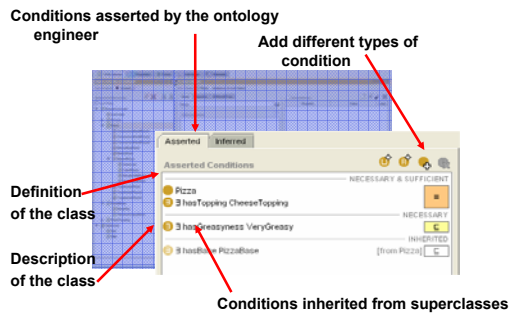
- Any Pizza that has some Cheese_Topping is a Cheesey_pizza.



Copyright Stanford University 2006

150

Conditions Widget



Copyright Stanford University 2006

151

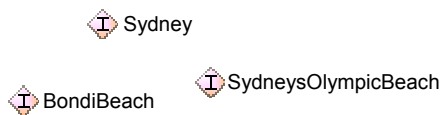
INDIVIDUALS

Copyright Stanford University 2006

152

Individuals

- Represent objects in the domain
- Specific things
- Two names could represent the same “real-world” individual



Copyright Stanford University 2006

153

Create instances



Create an instance of a class

- The class becomes a **direct type** of the instance
- Any superclass of the direct type is a **type** of the instance
- Generally, you create instances if you have a “type-of” something

Copyright Stanford University 2006

154

Outline

- Background on OWL
- Basic Protégé-OWL Usage
- ➔ • Classification
- Visualization components

Copyright Stanford University 2006

155

Reasoners

- Reasoners (“classifiers”) infer information that is not explicitly contained within the ontology
- Standard reasoner services are:
 - **Consistency Checking** (i.e., satisfiability—can a class have any instances?)
 - **Subsumption Checking** (Finding subclasses—is A a subclass of B?)
 - **Equivalence Checking**
 - **Instantiation Checking** (Which classes does an individual belong to)
- For Protégé we recommend RACER or Fact++ (but other tools with DIG support work too)
- Reasoners can be used at runtime in applications as a querying mechanism
- Used during development as an ontology “**compiler**”. Ontologies can be compiled to check if the meaning is what was intended

Copyright Stanford University 2006

156

Examples of DL Reasoning

- Detecting inconsistencies
 - VegetableAndMeat topping is a subclass of both Vegetable_Topping and Meat_Topping classes
 - But Vegetable_Topping and Meat_Topping are disjoint!
 - Classifier will report inconsistency
- Classify individuals
 - Pizza001 has a topping that is an instance of Cheesey_Topping
 - Therefore Pizza001 is an instance of Cheesey_Pizza

Copyright Stanford University 2006

157

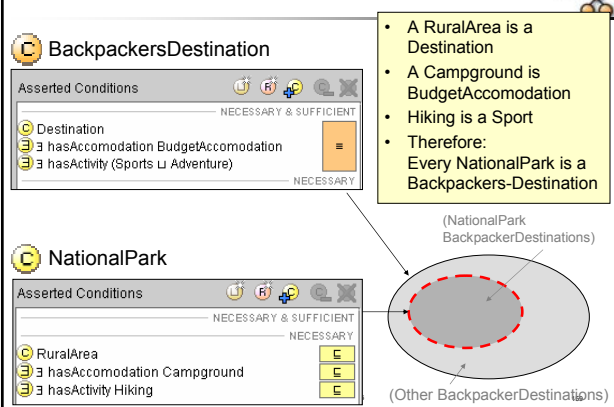
Open vs. Closed World reasoning

- Open world reasoning
 - Negation as contradiction
 - Anything might be true unless it can be proven false
 - Reasoning about *any world consistent with this one*
- Closed world reasoning
 - Negation as failure
 - Anything that cannot be found is false
 - Reasoning about *this world*

Copyright Stanford University 2006

158

Classification



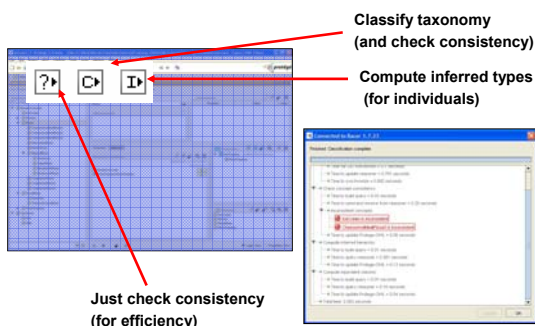
Run a DL Reasoner with Protégé OWL

- Protégé OWL can work with multiple reasoners
 - Racer (<http://www.racer-systems.com/>)
 - Pellet (<http://www.mindswap.org/2003/pellet/>)
 - Fact++ (<http://owl.man.ac.uk/factplusplus/>)
- Need to install, configure, and run at least one reasoner as a separate process
- Protégé OWL and reasoner exchange information through inter-process communication

Copyright Stanford University 2006

160

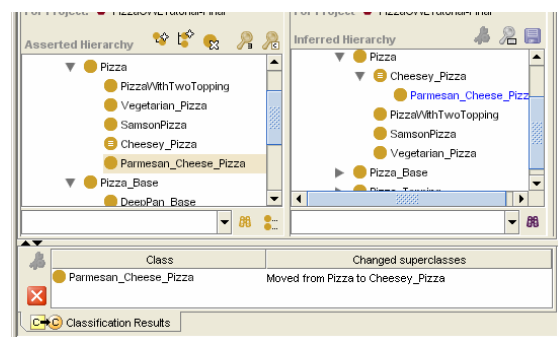
Accessing the Reasoner



Copyright Stanford University 2006

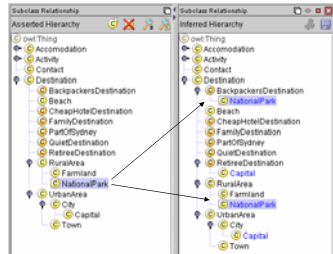
161

Result of Classification



Classification

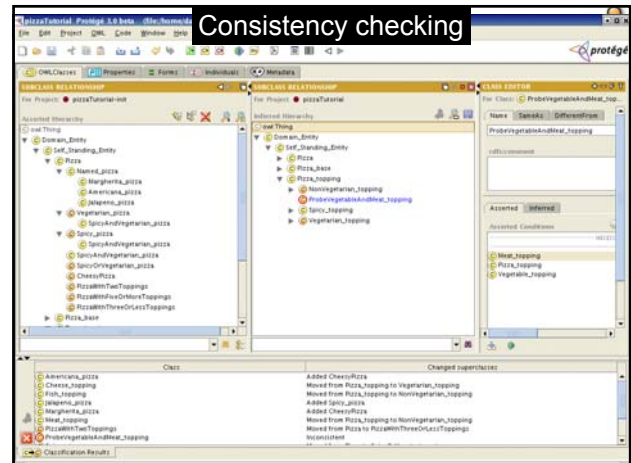
- Input: Asserted class definitions
- Output: Inferred subclass relationships



Copyright Stanford University 2006

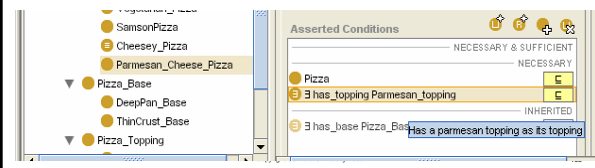
163

Consistency checking

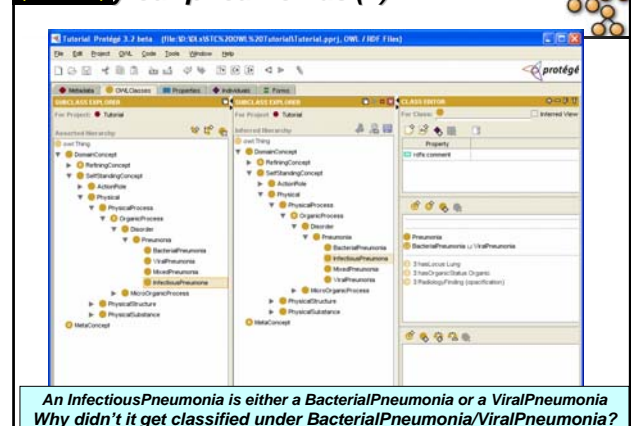


Subsumption Reasoning in OWL

- Defined classes allow a reasoner to determine that one class is necessarily the subclass of the defined class
- Example: Define *Parmesan_Cheese_Pizza* – A pizza that has some *Parmesan_topping*
Parmesan_Cheese_Pizza will be classified under *Cheesey_Pizza*

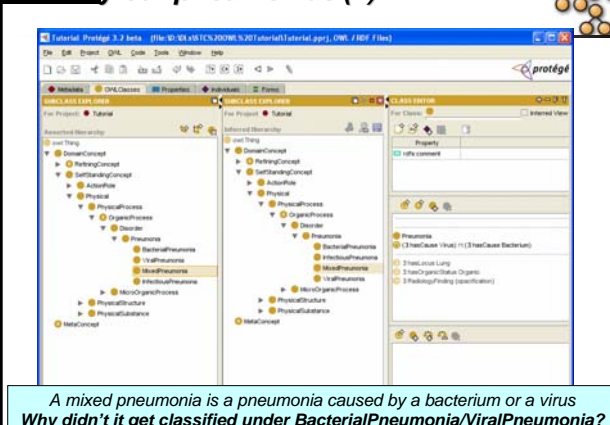


(DEMO) My our pneumonias (1)



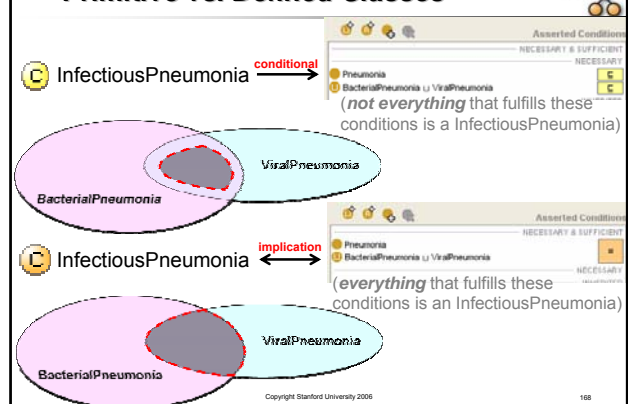
An *InfectiousPneumonia* is either a *BacterialPneumonia* or a *ViralPneumonia*
 Why didn't it get classified under *BacterialPneumonia/ViralPneumonia*?

(DEMO) My our pneumonias (2)



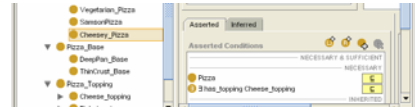
A *mixed pneumonia* is a *pneumonia* caused by a *bacterium* or a *virus*
 Why didn't it get classified under *BacterialPneumonia/ViralPneumonia*?

Primitive vs. Defined Classes

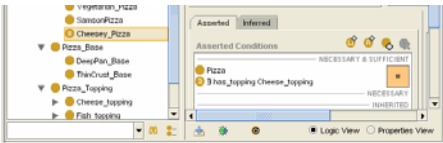


A Defined Class has Necessary AND Sufficient Conditions

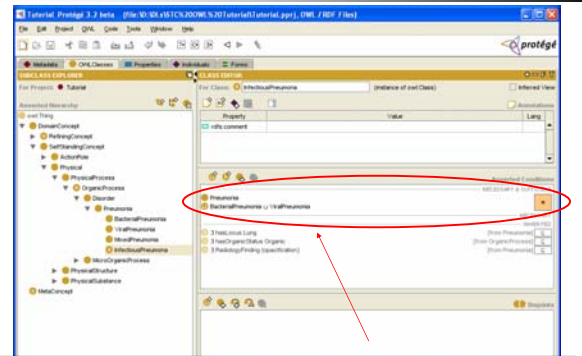
- Before conversion to a defined class



- After conversion to a defined class

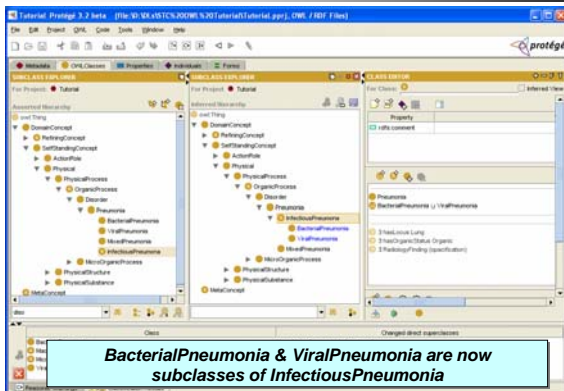


Make InfectiousPneumonia a defined class



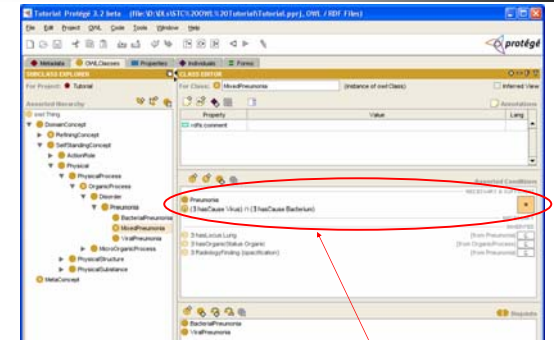
An infectious pneumonia is either a bacterial or viral pneumonia

Now classify...



BacterialPneumonia & ViralPneumonia are now subclasses of InfectiousPneumonia

Make MixedPneumonia, Bacterial, and ViralPneumonia defined classes



A mixed pneumonia is a pneumonia caused by a bacterium or a virus

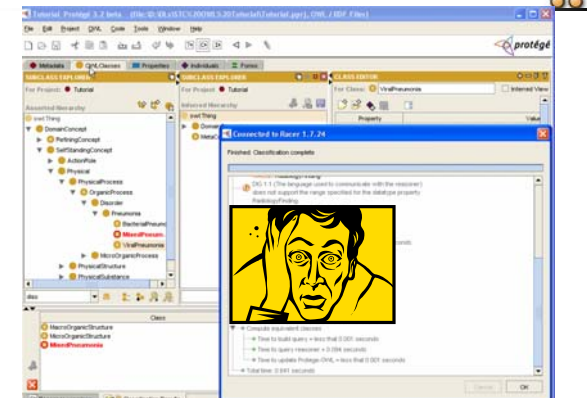
What it means

- Any** pneumonia caused by a bacterium is a bacterial pneumonia
- i.e., Something is a bacterial pneumonia **if and only if** it is a pneumonia and it is caused by a bacterium
- i.e., BacterialPneumonia \leftrightarrow Pneumonia AND causedBy some Bacterium
 - BacterialPneumonia \leftrightarrow Pneumonia $\wedge (\exists \text{ causedBy.Bacterium})$

Copyright Stanford University 2006

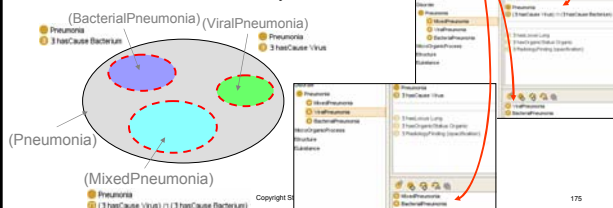
173

Classify our pneumonias...



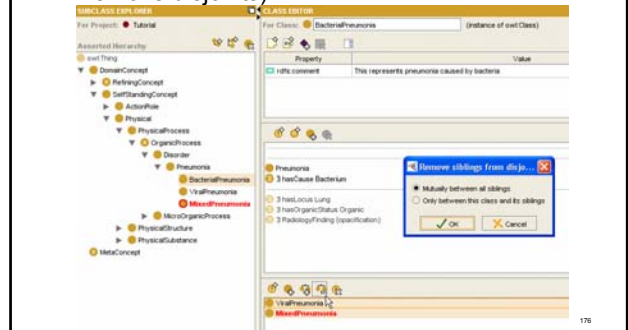
Why is MixedPneumonia inconsistent?

- MixedPneumonia comprises pneumonias caused by Virus and Bacteria
- Thus, it is inconsistent to say that BacterialPneumonia and ViralPneumonia are *disjoint*
 - MixedPneumonia asserts there exist pneumonias that hasCause Virus **and** hasCause Bacterium
 - Disjoints asserts that there are no individuals shared by BacterialPneumonia, ViralPneumonia, and MixedPneumonia!
- Solution: we must remove the disjoints constraint to resolve the inconsistency

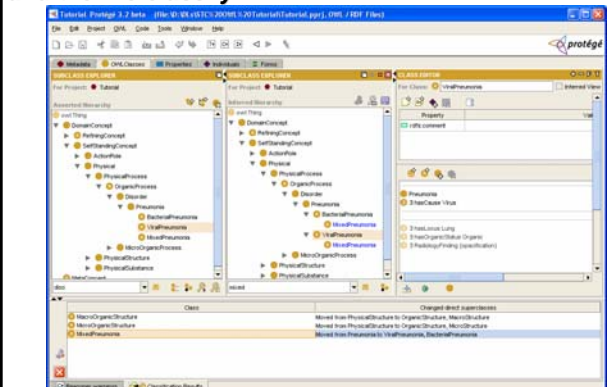


(DEMO) Remove disjoints

- (need to first make classes primitive to remove disjoints)

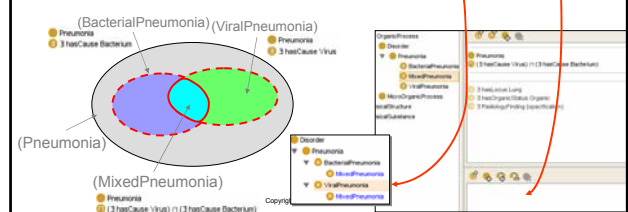


(DEMO) Make defined classes, and then re-classify...



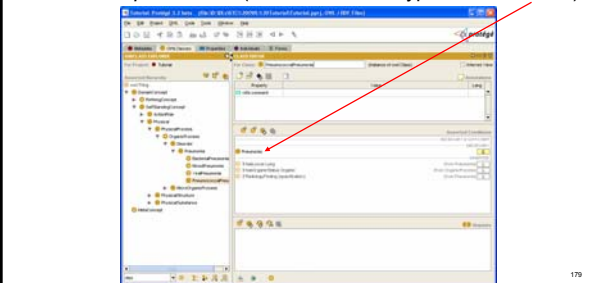
What does it mean?

- Bacterial, Viral, and MixedPneumonia are no longer disjoint
- MixedPneumonias are defined to be Pneumonias that hasCause Bacteria and Virus
- Thus, all MixedPneumonias are also both BacterialPneumonia and ViralPneumonia
 - MixedPneumonia is re-classified to have two parents

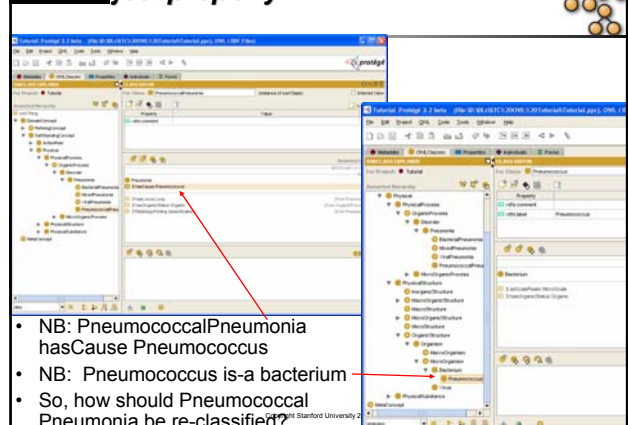


(DEMO) Complex inference based on object properties

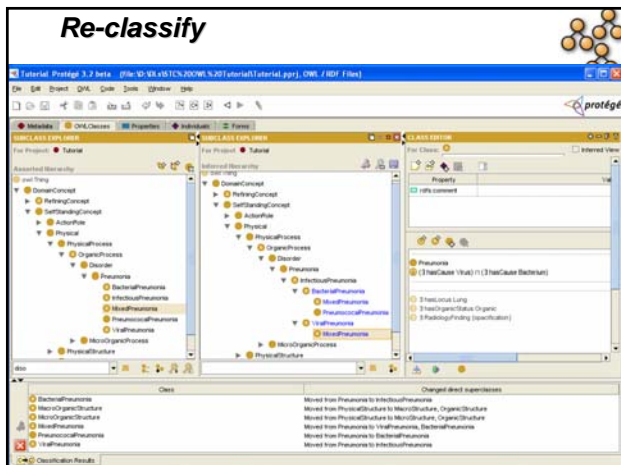
- Create **PneumococcalPneumonia**, which is caused by pneumococcus (a kind of bacteria)
- We will see if the classifier can recognize this is a *type of bacterial pneumonia* (so create it as a type of Pneumonia)



(DEMO) Object property



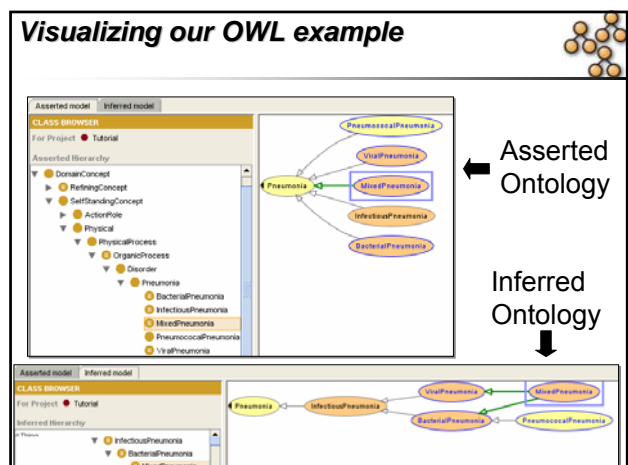
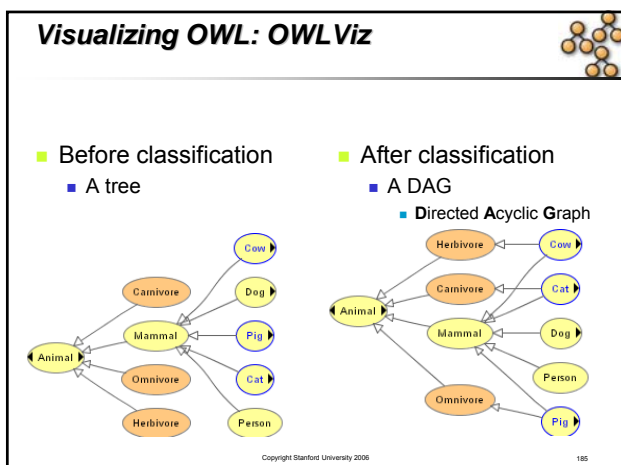
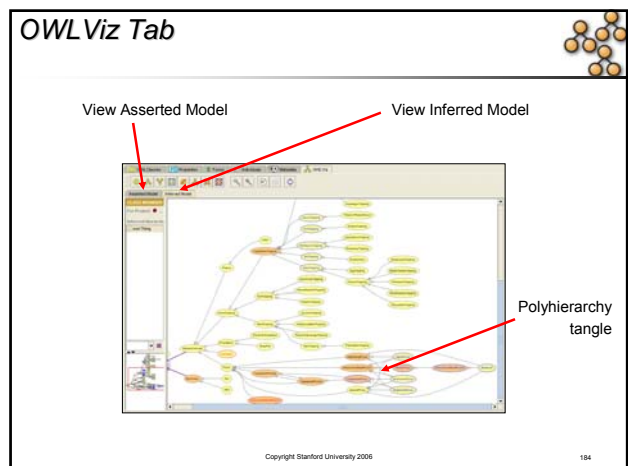
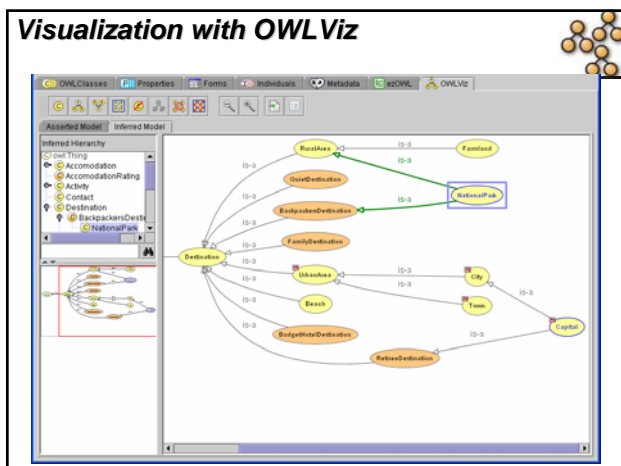
- NB: PneumococcalPneumonia hasCause Pneumococcus
- NB: Pneumococcus is-a bacterium
- So, how should PneumococcalPneumonia be re-classified?



Outline

- Background on OWL
- Basic Protégé-OWL Usage
- Classification
- ➔ • Visualization components

Copyright Stanford University 2006 182



What does all this mean?

- Description logic provides
 - Greater expressivity and semantic precision
 - Compositional definitions: define new concepts from old
 - Automatic classification & consistency checking
- New kinds of applications
 - Large terminology development
 - Semantic Web
- Protégé OWL provides a robust GUI environment for developing OWL ontologies

Copyright Stanford University 2006

187

Further reading/exploration

- Protégé: <http://protege.stanford.edu>
- Protégé OWL:
<http://protege.stanford.edu/plugins/owl/>
- OWL tutorial materials from CO-ODE project site (University of Manchester)
<http://www.co-ode.org/resources/tutorials/>
- CO-ODE/HyOntUse <http://www.co-ode.org/>
- Protégé Workshops (early 2006)
- Protégé International Conference
- Protégé OWL discussion list
- cBio (<http://bioontology.org>)

Copyright Stanford University 2006

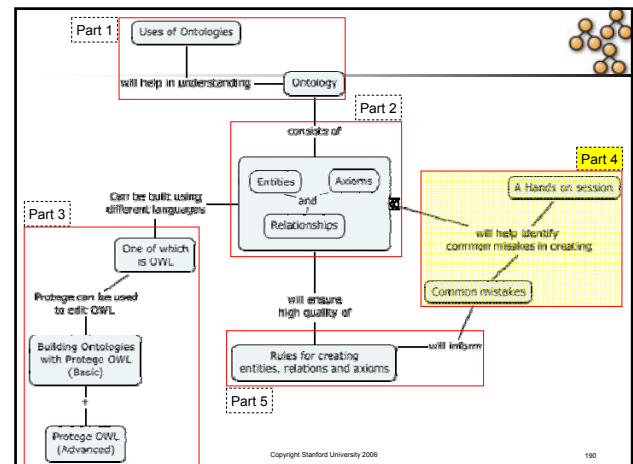
188

More about Protégé OWL

- Documentation on
<http://protege.stanford.edu/plugins/owl/documentation.html>
- Excellent tutorial by Mathew Horridge
<http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf>
- Other resources at <http://www.co-ode.org/resources/>

Copyright Stanford University 2006

189



Copyright Stanford University 2006

190

Hand on Session

Lets make an Ontology!

Exercise

- Goals
 - Create Ontology of Plants and Animals
- Steps
 1. Identify classes, properties, and instances
 2. Identify "definable" & "primitive" classes
 3. Organize primitive classes into a hierarchy
 4. Create relations between primitive classes using properties.
 5. Set domain and range constraints for the properties
 6. Define the "definable" things using primitives, properties and OWL axioms
 7. Check with Classifier

Copyright Stanford University 2006

192

Initial Terms

- Plant
- Lassie
- Animal
- Dog
- Cat
- Eats
- Cow
- Person
- Grass
- Herbivore
- Carnivore
- Gender
- Omnivore
- Buddha

Copyright Stanford University 2006

193

Common Mistakes

To much trust in natural language

- To much trust in natural language leads to ambiguities. E.g. 'ontology' is used systematically ambiguous in natural language in order to refer:
 - (a) to a field of scientific research and
 - (b) a type of certain artifacts that are created by researchers.
 -
- These are quite different entities that have to be treated as distinct entities.
- People tend to trust natural language naively and assume the following correspondence:
 - One natural language expression corresponds to one entity.

Copyright Stanford University 2006

195

Naive conceptualizations

- Most computer scientists embrace naive conceptualization, they declare things like
 - 'Fake Diamond is_a Diamond'
 - 'Absent leg is_a leg'.
 - Besides the fact that it is nonsense, this is wrong, because now 'Absent leg' will inherit all properties from 'leg'.

Copyright Stanford University 2006

196

Logical ambiguity

- In the gene ontology "x part_of y" sometimes means
 - "for any x that is an instance of X, there is a y that is an instance of Y such that x is part of y".
 - Sometimes it means "for some x that are instances of X, there is a y that is an instance of Y such that x is part of y".

Some-Some STRUCTURE

Copyright Stanford University 2006

197

Confusion caused by "is_a"

- Use of "is_a" as a technical term that covers the **instance_of** relation and the **subtype** relation.
- This leads to confusion, if we want to speak about a type of types. (Types that have types as instances, not as subtypes.)
 - E.g. the type red (= the type of red entities) is a subtype of the type of colored (= the type of all colored entities).
 - Further, the type red is an instance of the type of all colors (not of the type colored).
- This red book is an instance of the type colored, but not an instance of color. In contrast, red is not an instance of the type colored, but is an instance of the type color.

Copyright Stanford University 2006

198

Test case for is_a overloading

- The subtype relation is such that any property that is linked to a parent node in a is_a hierarchy is inherited by the child nodes. People tend to ignore that and to use is_a, whenever they have a relation that leads to a tree-like structure.
- For example, if you create an ontology that involves the following terms: "Soldier, officer, general, enlisted rank, private, colonel, lieutenant, U.S. army employee, captain, admiral, sergeant".
- I bet, that at least some of you will come up with a hierarchy of ranks with general on top and sergeant on the bottom of the hierarchy
 - ...and won't notice that you used is_a in order to express the relation is_superior_to.

Copyright Stanford University 2006

199

Too much information in one ontology

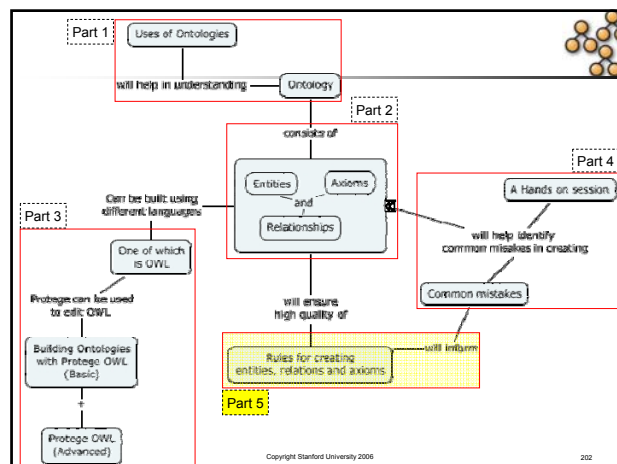
- Most so-called ontologies are basically is_a hierarchies of substance particulars. (Examples are the taxonomy of biological species or anatomical ontologies.)
- In these cases the ontologies consist of types that are essential to the entities that are instances of the types.
- People often make the mistake to include information in the ontology that is relevant to them, but does not belong there; like information about development state or whether something is pathological.
- It starts innocent -- and suddenly there are types like 'brown, unhappy and 6 year old dog' in your ontology.
- The right solution is to keep the ontology of substance particulars and the ontology of attributes distinct.

Copyright Stanford University 2006

200

ICD10 (1999): 587 codes for such accidents

- V31.22 Occupant of three-wheeled motor vehicle injured in collision with pedal cycle, person on outside of vehicle, nontraffic accident, while working for income**
- W65.40 Drowning and submersion while in bath-tub, street and highway, while engaged in sports activity**
- X35.44 Victim of volcanic eruption, street and highway, while resting, sleeping, eating or engaging in other vital activities**



Copyright Stanford University 2006

202

Do's and Don'ts while creating your Ontology

Based on work of Barry Smith

Why do we need [a higher] guidance?

- Ontologies must be intelligible both to humans (for annotation) and to machines (for reasoning and error-checking)
- Unintuitive rules for classification lead to entry errors (problematic links)
- Facilitate training of curators
- Overcome obstacles to mapping with other ontology and terminology systems
- Enhance harvesting of content through automatic reasoning systems

Copyright Stanford University 2006

204

First Commandment: Univocity

- Terms (including those describing relations) should have the **same meaning on every occasion** of use.
- In other words, they should refer to the same kinds of entities in reality
- Problem example: 'chromosome' in Sequence Ontology and in Cell Component Ontology means different things

Copyright Stanford University 2006

205

Example of univocity problem

(Old) Gene Ontology:

- 'part_of' = 'may be part of'
 - flagellum part_of cell
- 'part_of' = 'is at times part of'
 - replication fork part_of the nucleoplasm
- 'part_of' = 'is included as a sub-list in'

Copyright Stanford University 2006

206

Second Commandment: Positivity

- Complements of classes are not themselves classes.
- Terms such as 'non-mammal' or 'non-membrane' do not designate genuine classes.

Copyright Stanford University 2006

207

Third Commandment: Objectivity

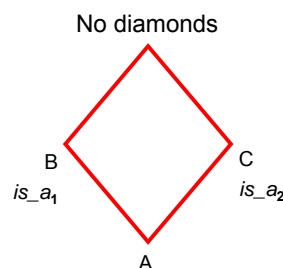
- Which classes exist is not a function of our biological knowledge.
- Terms such as 'unknown' or 'unclassified' or 'unlocalized':
 - do not designate biological natural kinds
 - do not designate differentiating characteristics [differential] of biological natural kinds

Copyright Stanford University 2006

208

Fourth Commandment: Single Inheritance

No class in a classification hierarchy should have more than one is_a parent on the immediate higher level

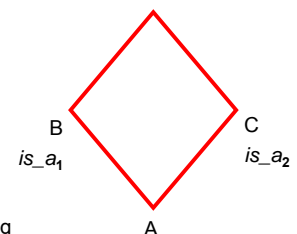


Copyright Stanford University 2006

209

Problems with multiple inheritance

- 'is_a' has two meanings – breaks the rule of univocity
- the multiple meanings makes coherent integration across ontologies difficult
- Benefit: keeps the ontology simple by having multiple sorts of partitions brought together within the same framework



Copyright Stanford University 2006

210

Definitions should be intelligible to both machines and humans

- Machines can cope with the full formal representation
- Humans need to use modularity
- **Plasma membrane**
 - *is a cell part* [immediate parent]
 - that *surrounds* the **cytoplasm** [differentia]

Copyright Stanford University 2006

217

Principle of Compositionality

- The meanings of compound terms should be determined by
 - the meanings of component terms
 - together with the rules governing syntax

Copyright Stanford University 2006

218

Principle of Syntactic Separateness

- Do not confuse sentences with ontology terms
- If you want to say: No As are Bs
 - do not invent a new class of non-Bs and say A **is_a** non-B

Copyright Stanford University 2006

219

Keep Epistemology Separate

- If you want to say that we do not know where As are located do not invent a new class of A's with unknown locations
 - Example: Holliday junction helicase complex **is-a** unlocalized
- A well-constructed ontology should grow linearly [monotonically];
 - it should not need to delete classes or relations because of increases in knowledge

Copyright Stanford University 2006

220

Some other rules of thumb

1. Don't confuse entities with concepts
2. Don't confuse entities with ways of getting to know entities
 - a brain is not the same as its CT-scan
3. Don't confuse entities with ways of talking about entities
 - A person's medical record is not == person himself
4. Don't confuse entities with artifacts of your database representation ...
 - e.g. multiple dosing event in PharmGKB
5. An ontology should not change when the ontology language changes
 - The process of driving a car doesn't change whether you describe it in English or Spanish.

Copyright Stanford University 2006

221

Guidelines for instances

- Every class has at least one instance
- Each child class has a smaller set of instances than its parent class
- Distinct classes on the same level never share instances
- Distinct leaf classes within a classification never share instances

Copyright Stanford University 2006

222

Principles for defining relations in ontologies

Biomedical ontology integration / interoperability

- Will not be achieved through integration of meanings or concepts
- The problem is precisely that different user communities use *different concepts*
- ***What's really needed is to have well-defined commonly used relationships***

Ontologically rigorous relations

- Move from associative relations between meanings/concepts to strictly defined [ontological] relations between the entities themselves.
- It is not enough to consider just classes or types.
 - We need also to take account of *instances* and *time*
- The relations can then be used computationally

Benefits of well-defined relationships

- If the relations in an ontology are well-defined [All-Some structure], then reasoning can cascade from one relational assertion ($A R_1 B$) to the next ($B R_2 C$).
- Relations used in ontologies thus far have not been well defined in this sense.
- *Find all DNA binding proteins* should also find all transcription factor proteins because
 - *Transcription factor is_a DNA binding protein*

An unclear definition of *is_a*

- 'A' is more specific in meaning than 'B'
- HL7-RIM:
 - Individual Allele *is_a* Act of Observation
 - cancer documentation *is_a* cancer
 - disease prevention *is_a* disease

How to define the *is_a* relation

- *What does $A \text{ is_a } B$ mean?*
- For all x , if x **instance_of** A then x **instance_of** some B
- *cell division **is_a** biological process*

ALL-SOME STRUCTURE

An unclear definition of part_of

A *part_of* B:

A composes (with one or more other physical units) some larger whole B

This confuses relations between meanings or concepts with relations entities in reality

Copyright Stanford University 2006

229

How to define A *part_of* B

- What does A *part_of* B mean?
- For all x, if x *instance_of* A then there is some y, y *instance_of* B and x *part_of* y
 - where '*part_of*' is the instance-level part relation
- *cell nucleus part_of cell*

ALL-SOME STRUCTURE

Copyright Stanford University 2006

230

Kinds of relations

- Between classes:
 - *is_a*, *part_of*, ...
- Between an instance and a class
 - this explosion *instance_of* the class explosion
- Between instances:
 - Mary's heart *part_of* Mary

Copyright Stanford University 2006

231

How many relations do we need?

Properties of Relations

1. Transitivity
2. Symmetry
3. Reflexivity
4. Anti-Symmetry
5. ...

- Avoid putting '*_*' between arbitrary characters and calling it a relation
- *is_somewhat_related_to* is the worst kind of relation to create!

Copyright Stanford University 2006

232

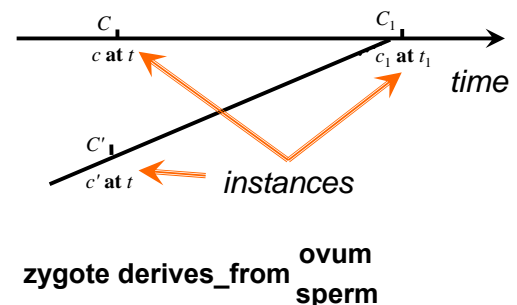
Don't forget instances when defining relations

- *part_of* as a relation between classes versus *part_of* as a relation between instances
 - *nucleus part_of cell*
 - your heart *part_of* you
- What holds on the level of instances may not hold on the level of universals
 - *nucleus adjacent_to cytoplasm*
 - **Not:** *cytoplasm adjacent_to nucleus*
 - *seminal vesicle adjacent_to urinary bladder*
 - **Not:** *urinary bladder adjacent_to seminal vesicle*

Copyright Stanford University 2006

233

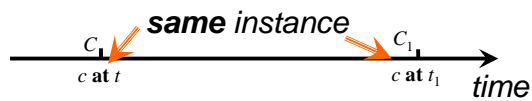
Time matters ... e.g. *derives_from*



Copyright Stanford University 2006

234

transformation_of



pre-RNA → mature RNA
child → adult

Copyright Stanford University 2006

235

The “take home”

- Follow a methodology which enforces **clear, coherent definitions for entities and relationships**
- This promotes quality assurance
 - intent is not hard-coded into software
 - Meaning of relationships is defined, not inferred
- **Enables automated reasoning** across ontologies and across data at different granularities

Copyright Stanford University 2006

236

Acknowledgements

- NCBO is funded by NIH Roadmap initiative
- Protégé and Protégé-OWL are supported by grants and contracts from the NIH

Copyright Stanford University 2006

237

Questions?

Protégé:

<http://protege.stanford.edu>

NCBO:

<http://bioontology.org>

End of presentation

